

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-235519

(43)Date of publication of application : 29.08.2000

(51)Int.Cl. G06F 12/08  
G06F 12/12

(21)Application number : 11-356678

(71)Applicant : NEC CORP

(22)Date of filing : 15.12.1999

(72)Inventor : MIYAZAKI MITSUHIRO

(30)Priority

Priority number : 10356815 Priority date : 15.12.1998 Priority country : JP

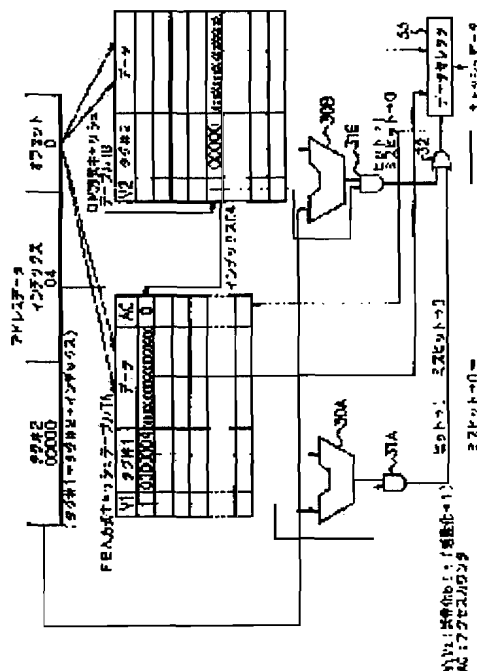
## (54) CACHE SYSTEM AND CACHE PROCESSING METHOD

(57)Abstract:

PROBLEM TO BE SOLVED: To realize both of high speed data access and a high cache bit rate at high levels by improving the storing performance of a cache line having a high bit rate and the using efficiency of a cache memory space.

SOLUTION: In a cache part obtained by combining an FSA type cache and a cache such as a DM type cache other than the FSA type, tag comparison is simultaneously executed by both the caches. When count values related to the numbers of cache hits of respective cache lines of the FSA type cache are stored and a certain cache line in the FSA type e.g. is hit, '1' is added to the count value of the cache line, and when both cache tables are mis-hit, '1' is simultaneously subtracted from the count values of all cache lines.

Since the data of respective caches are managed/transferred by using the count values in accordance with respective cases, the overlap of data in both the caches is removed, a cache line of a high bit rate is stored in the FSA type cache and the FSA type cache of a high speed and a high bit rate can be realized.



## LEGAL STATUS

[Date of request for examination] 15.12.1999

[Date of sending the examiner's decision of rejection] 30.07.2002

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision  
of rejection]

[Date of requesting appeal against examiner's  
decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-235519

(P2000-235519A)

(43) 公開日 平成12年8月29日 (2000.8.29)

(51) IntCl. <sup>7</sup>	識別記号	F I	テ-リ-ト* (参考)
G 0 6 F 12/08		G 0 6 F 12/08	F
			B
			S
12/12		12/12	A

審査請求 有 請求項の数27 O L (全 29 頁)

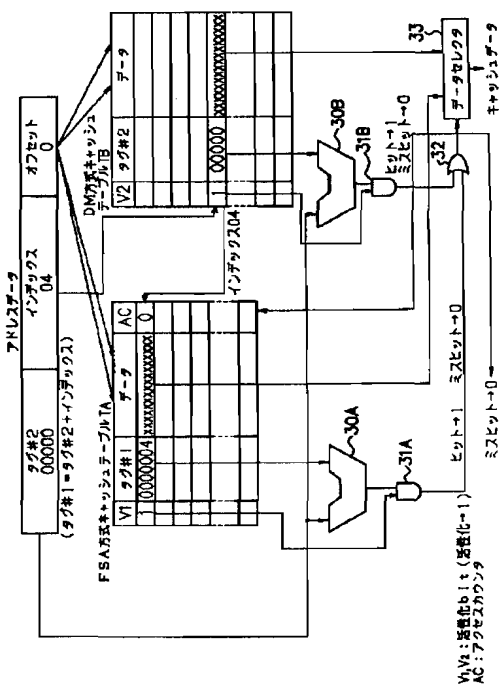
(21) 出願番号	特願平11-356678	(71) 出願人	000004237 日本電気株式会社 東京都港区芝五丁目7番1号
(22) 出願日	平成11年12月15日 (1999. 12. 15)	(72) 発明者	宮崎 充弘 東京都港区芝五丁目7番1号 日本電気株式 会社社内
(31) 優先権主張番号	特願平10-356815	(74) 代理人	100084250 弁理士 丸山 隆夫
(32) 優先日	平成10年12月15日 (1998. 12. 15)		
(33) 優先権主張国	日本 (J P)		

(54) 【発明の名称】 キャッシュシステムおよびキャッシュ処理方法

(57) 【要約】

【課題】 使用頻度の高いキャッシュラインの保持性能とキャッシュメモリ空間の使用効率が低く、ヒット率がまだ低かった。

【解決手段】 FSA方式キャッシュとDM方式キャッシュ等のFSA方式以外のキャッシュとを組み合わせたキャッシュ部においてタグ比較が両キャッシュで同時に行われる。FSA方式キャッシュの各キャッシュラインのキャッシュヒットの数に関するカウント値が保持され、例えばFSA方式キャッシュのあるキャッシュラインがヒットした場合にはそのキャッシュラインのカウント値が+1され、両キャッシュテーブルがミスヒットした場合には全キャッシュラインのカウント値が一斉に-1される。このカウント値を用いて各場合に依じて各キャッシュのデータの管理/転送が行われることにより、両キャッシュへのデータの重複が無くされ、FSA方式キャッシュに高ヒット率のキャッシュラインが保持され、高速かつ高ヒット率の前記キャッシュ部が実現される。



## 【特許請求の範囲】

【請求項1】 CPU (Central Processing Unit) (50)、MPU (Micro Processor Unit) 等のデータ処理ユニット (50) の、メインメモリ (21) 等の外部メモリ (20, 21) へのアクセス回数を削減するためのキャッシュシステムにおいて、

キャッシュヒット率の高いキャッシュラインの保持が可能なFSA (フルセットアソシアティブ) 方式キャッシュ (TA) であり、入力アドレスデータから抽出された第1のタグ (#1) が前記FSA方式キャッシュ (TA) の活性なキャッシュラインに格納されていた第1のタグ (#1) の一つと一致した場合に、その一致した第1のタグ (#1) に対応する活性なキャッシュラインに格納されていたデータが前記データ処理ユニット (50) によりキャッシュデータとして読み込まれるFSA方式キャッシュ (TA) と、

前記キャッシュシステムにおいて前記FSA方式キャッシュ (TA) と組み合わせられる形で設けられキャッシュヒット判定のためのタグ比較を前記FSA方式キャッシュ (TA) と同時に行うFSA方式以外のキャッシュ (TB, TC) であり、前記入力アドレスデータから抽出された第2のタグ (#2) が前記FSA方式以外のキャッシュ (TB, TC) の前記入力アドレスデータから抽出されたインデックスに対応する活性なキャッシュラインに格納されていた第2のタグ (#2) に一致した場合に、その一致した第2のタグ (#2) に対応する活性なキャッシュラインに格納されていたデータが前記データ処理ユニット (50) によりキャッシュデータとして読み込まれるFSA方式以外のキャッシュ (TB, TC) と、

前記FSA方式キャッシュ (TA) の各キャッシュラインに対応して設けられ、前記キャッシュラインに生じたキャッシュヒットの回数に関するキャッシュヒットカウント値を格納するキャッシュヒットカウント格納手段 (AC) と、

前記FSA方式キャッシュ (TA) の各キャッシュラインのキャッシュヒットをカウントし、前記キャッシュヒットカウント格納手段 (AC) に格納される前記キャッシュヒットカウント値を管理し更新するキャッシュヒットカウント管理手段とを備え、

前記FSA方式以外のキャッシュ (TB, TC) に前記インデックスに対応する非活性なキャッシュラインが存在せず前記FSA方式キャッシュ (TA) に1以上の非活性なキャッシュラインが存在する時に前記FSA方式キャッシュ (TA) と前記FSA方式以外のキャッシュ (TB, TC) の両方がミスヒットした場合は、前記FSA方式以外のキャッシュ (TB, TC) のミスヒットした前記インデックスに対応するあるキャッシュラインに格納されていたデータが前記FSA方式キャッシュ

(TA) の前記非活性なキャッシュラインのうちの一つに転送され、前記両方のキャッシュのミスヒットにより前記外部メモリ (20, 21) からフェッチされたデータは前記FSA方式キャッシュ (TA) へのデータ転送元となった前記FSA方式以外のキャッシュ (TB, TC) の前記キャッシュラインに書き込まれ、

前記FSA方式以外のキャッシュ (TB, TC) に前記インデックスに対応する非活性なキャッシュラインが存在せず前記FSA方式キャッシュ (TA) が活性なキャッシュラインで一杯な時に前記FSA方式キャッシュ

(TA) と前記FSA方式以外のキャッシュ (TB, TC) の両方がミスヒットした場合は、前記FSA方式以外のキャッシュ (TB, TC) のミスヒットした前記インデックスに対応するあるキャッシュラインに格納されていたデータが前記FSA方式キャッシュ (TA) の最小のキャッシュヒットカウント値を持つキャッシュラインのうちの一つに転送され、前記両方のキャッシュのミスヒットにより前記外部メモリ (20, 21) からフェッチされたデータは前記FSA方式キャッシュ (TA) へのデータ転送元となった前記FSA方式以外のキャッシュ (TB, TC) の前記キャッシュラインに書き込まれ、

前記FSA方式以外のキャッシュ (TB, TC) に前記インデックスに対応する非活性なキャッシュラインが存在する時に前記FSA方式キャッシュ (TA) と前記FSA方式以外のキャッシュ (TB, TC) の両方がミスヒットした場合は、前記両方のキャッシュのミスヒットにより前記外部メモリ (20, 21) からフェッチされたデータは前記FSA方式以外のキャッシュ (TB, TC) の前記インデックスに対応する非活性なキャッシュラインに書き込まれることを特徴とするキャッシュシステム。

【請求項2】 前記FSA方式キャッシュ (TA) のあるキャッシュラインがキャッシュヒットした場合には、前記キャッシュヒットカウント管理手段はそのキャッシュラインの前記キャッシュヒットカウント値を1つインクリメントし、

前記FSA方式キャッシュ (TA) と前記FSA方式以外のキャッシュ (TB, TC) の両方がミスヒットした場合には、前記キャッシュヒットカウント管理手段は前記FSA方式キャッシュ (TA) の全キャッシュラインの前記キャッシュヒットカウント値を一斉に1つデクリメントすることを特徴とする請求項1記載のキャッシュシステム。

【請求項3】 前記FSA方式キャッシュ (TA) のあるキャッシュラインがキャッシュヒットした場合には、前記キャッシュヒットカウント管理手段はそのキャッシュラインの前記キャッシュヒットカウント値を1つインクリメントし、

前記FSA方式キャッシュ (TA) がミスヒットした場

10

20

30

40

50

合には、前記キャッシュヒットカウント管理手段は前記 FSA 方式キャッシュ (TA) の全キャッシュラインの前記キャッシュヒットカウント値を一斉に 1 つデクリメントすることを特徴とする請求項 1 記載のキャッシュシステム。

【請求項 4】 前記 FSA 方式キャッシュ (TA) のあるキャッシュラインがキャッシュヒットした場合には、前記キャッシュヒットカウント管理手段はそのキャッシュラインの前記キャッシュヒットカウント値を 1 つインクリメントすることを特徴とする請求項 1 記載のキャッシュシステム。

【請求項 5】 前記 FSA 方式以外のキャッシュ (TB, TC) として、ダイレクトマッピング方式キャッシュ (TB) が用いられることを特徴とする請求項 1 から 4 のうちのいずれか 1 項に記載のキャッシュシステム。

【請求項 6】 前記 FSA 方式以外のキャッシュ (TB, TC) として、Nウェイセットアソシアティブキャッシュ (TC) (N=2, 4, 8, ...) が用いられることを特徴とする請求項 1 から 4 のうちのいずれか 1 項に記載のキャッシュシステム。

【請求項 7】 前記 FSA 方式キャッシュ (TA) の各キャッシュラインに対応して設けられ、前記キャッシュラインに生じた最新のキャッシュヒットの日時に関するデータを格納するキャッシュヒット日時格納手段をさらに備え、

前記 FSA 方式以外のキャッシュ (TB, TC) に前記インデックスに対応する非活性なキャッシュラインが存在せず前記 FSA 方式キャッシュ (TA) が活性なキャッシュラインで一杯な時に前記 FSA 方式キャッシュ

(TA) と前記 FSA 方式以外のキャッシュ (TB, TC) の両方がミスヒットした前記場合には、前記キャッシュヒット日時格納手段に格納された前記データに基づいて前記 FSA 方式キャッシュ (TA) の前記最小のキャッシュヒットカウント値を持つ前記キャッシュラインの中から最近のキャッシュヒットが最も古いキャッシュラインが選択され、その選択されたキャッシュラインが前記 FSA 方式以外のキャッシュ (TB, TC) から前記 FSA 方式キャッシュ (TA) への前記データ転送の転送先として指定されることを特徴とする請求項 1 から 6 のうちのいずれか 1 項に記載のキャッシュシステム。

【請求項 8】 各インデックスに対応して設けられ、そのインデックスに対応する N 本のキャッシュラインの中で最近のキャッシュヒットが最も古いキャッシュラインを示すための LRU (Least Recently Used) 値を格納する LRU 格納手段をさらに備え、前記 Nウェイセットアソシアティブキャッシュ (TC) に前記インデックスに対応する非活性なキャッシュラインが存在せず前記 FSA 方式キャッシュ (TA) に 1 以上の非活性なキャッシュラインが存在する時に前記 FSA 方式キャッシュ (TA) と前記 Nウェイセットアソシ

アティブキャッシュ (TC) の両方がミスヒットした前記場合には、前記 Nウェイセットアソシアティブキャッシュ (TC) の N 本のミスヒットしたキャッシュラインの中から前記 LRU 値により指定されたキャッシュラインが選択され、その選択されたキャッシュラインに格納されていたデータが前記 FSA 方式キャッシュ (TA) の前記非活性なキャッシュラインのうちの一つに転送され、前記両方のキャッシュのミスヒットにより前記外部メモリ (20, 21) からフェッチされたデータは前記 FSA 方式キャッシュ (TA) へのデータ転送元となった前記 Nウェイセットアソシアティブキャッシュ (TC) の前記選択されたキャッシュラインに書き込まれ、前記 Nウェイセットアソシアティブキャッシュ (TC) に前記インデックスに対応する非活性なキャッシュラインが存在せず前記 FSA 方式キャッシュ (TA) が活性なキャッシュラインで一杯な時に前記 FSA 方式キャッシュ (TA) と前記 Nウェイセットアソシアティブキャッシュ (TC) の両方がミスヒットした前記場合には、前記 Nウェイセットアソシアティブキャッシュ (TC) の N 本のミスヒットしたキャッシュラインの中から前記 LRU 値により指定されたキャッシュラインが選択され、その選択されたキャッシュラインに格納されていたデータが前記 FSA 方式キャッシュ (TA) の前記最小のキャッシュヒットカウント値を持つ前記キャッシュラインのうちの一つに転送され、前記両方のキャッシュのミスヒットにより前記外部メモリ (20, 21) からフェッチされたデータは前記 FSA 方式キャッシュ (TA) へのデータ転送元となった前記 Nウェイセットアソシアティブキャッシュ (TC) の前記選択されたキャッシュラインに書き込まれ、

前記 Nウェイセットアソシアティブキャッシュ (TC) に前記インデックスに対応する非活性なキャッシュラインが 1 以上存在する時に前記 FSA 方式キャッシュ (TA) と前記 Nウェイセットアソシアティブキャッシュ (TC) の両方がミスヒットした前記場合には、前記両方のキャッシュのミスヒットにより前記外部メモリ (20, 21) からフェッチされたデータは前記 Nウェイセットアソシアティブキャッシュ (TC) の前記インデックスに対応する前記非活性なキャッシュラインのうちの一つのキャッシュラインに書き込まれることを特徴とする請求項 6 記載のキャッシュシステム。

【請求項 9】 前記 FSA 方式キャッシュ (TA) と前記 FSA 方式以外のキャッシュ (TB, TC) からなる一次キャッシュ (15) に付加して設けられた二次キャッシュ (20) をさらに備えることを特徴とする請求項 1 から 8 のうちのいずれか 1 項に記載のキャッシュシステム。

【請求項 10】 キャッシュヒット率の高いキャッシュラインの保持が可能な FSA (フルセットアソシアティブ) 方式キャッシュ (TA) と前記 FSA 方式キャッシュ

10

20

30

40

50

ユ (TA) と同時にキャッシュヒット判定のためのタグ比較を行うFSA方式以外のキャッシュ (TB, TC) との組み合わせを用い、CPU (Central Processing Unit) (50)、MPU (MicroProcessor Unit) 等のデータ処理ユニット (50) の、メインメモリ (21) 等の外部メモリ (20, 21) へのアクセス回数を削減するためのキャッシュ処理方法において、

入力アドレスデータから抽出された第1のタグ (#1) が前記FSA方式キャッシュ (TA) の活性なキャッシュラインに格納されていた複数の第1のタグ (#1) と比較され、前記第1のタグ (#1) の一致が有った場合に前記FSA方式キャッシュ (TA) がキャッシュヒットしたと判定される第1のタグ比較工程と、

前記第1のタグ比較工程と同時に進められ、前記入力アドレスデータから抽出された第2のタグ (#2) が前記FSA方式以外のキャッシュ (TB, TC) の前記入力アドレスデータから抽出されたインデックスに対応する1以上の活性なキャッシュラインに格納されていた第2のタグ (#2) と比較され、前記第2のタグ (#2) の一致が有った場合に前記FSA方式以外のキャッシュ (TB, TC) がキャッシュヒットしたと判定される第2のタグ比較工程と、

前記FSA方式キャッシュ (TA) の各キャッシュラインに起きたキャッシュヒットがカウントされ、前記FSA方式キャッシュ (TA) の各キャッシュラインに対応して設けられたキャッシュヒットカウント格納手段 (AC) に格納された各キャッシュライン毎のキャッシュヒットカウント値が管理し更新されるキャッシュヒットカウント管理工程と、

前記第1のタグ比較工程において前記FSA方式キャッシュ (TA) がキャッシュヒットした場合に、前記FSA方式キャッシュ (TA) の前記一致した第1のタグ (#1) に対応するキャッシュラインに格納されていたデータが、前記データ処理ユニット (50) によりキャッシュデータとして読み込まれる第1のキャッシュデータ読み込み工程と、

前記第2のタグ比較工程において前記FSA方式以外のキャッシュ (TB, TC) がキャッシュヒットした場合に、前記FSA方式以外のキャッシュ (TB, TC) の前記インデックスと前記一致した第2のタグ (#2) に対応するキャッシュラインに格納されていたデータが、前記データ処理ユニット (50) によりキャッシュデータとして読み込まれる第2のキャッシュデータ読み込み工程と、

前記FSA方式以外のキャッシュ (TB, TC) に前記インデックスに対応する非活性なキャッシュラインが存在せず前記FSA方式キャッシュ (TA) に1以上の非活性なキャッシュラインが存在する時に前記FSA方式キャッシュ (TA) と前記FSA方式以外のキャッシュ

(TB, TC) の両方がミスヒットした場合に、前記FSA方式以外のキャッシュ (TB, TC) のミスヒットした前記インデックスに対応するあるキャッシュラインに格納されていたデータが前記FSA方式キャッシュ

(TA) の前記非活性なキャッシュラインのうちの一つに転送され、前記両方のキャッシュのミスヒットにより前記外部メモリ (20, 21) からフェッチされたデータが前記FSA方式キャッシュ (TA) へのデータ転送元となった前記FSA方式以外のキャッシュ (TB, TC) の前記キャッシュラインに書き込まれる第1のデータ転送工程と、

前記FSA方式以外のキャッシュ (TB, TC) に前記インデックスに対応する非活性なキャッシュラインが存在せず前記FSA方式キャッシュ (TA) が活性なキャッシュラインで一杯な時に前記FSA方式キャッシュ

(TA) と前記FSA方式以外のキャッシュ (TB, TC) の両方がミスヒットした場合に、前記FSA方式以外のキャッシュ (TB, TC) のミスヒットした前記インデックスに対応するあるキャッシュラインに格納されていたデータが前記FSA方式キャッシュ (TA) の最小のキャッシュヒットカウント値を持つキャッシュラインのうちの一つに転送され、前記両方のキャッシュのミスヒットにより前記外部メモリ (20, 21) からフェッチされたデータが前記FSA方式キャッシュ (TA) へのデータ転送元となった前記FSA方式以外のキャッシュ (TB, TC) の前記キャッシュラインに書き込まれる第2のデータ転送工程と、

前記FSA方式以外のキャッシュ (TB, TC) に前記インデックスに対応する非活性なキャッシュラインが存在する時に前記FSA方式キャッシュ (TA) と前記FSA方式以外のキャッシュ (TB, TC) の両方がミスヒットした場合に、前記両方のキャッシュのミスヒットにより前記外部メモリ (20, 21) からフェッチされたデータが前記FSA方式以外のキャッシュ (TB, TC) の前記インデックスに対応する非活性なキャッシュラインに書き込まれる第3のデータ転送工程とを備えることを特徴とするキャッシュ処理方法。

【請求項11】 前記キャッシュヒットカウント管理工程において、

前記FSA方式キャッシュ (TA) のあるキャッシュラインがキャッシュヒットした場合には、そのキャッシュラインの前記キャッシュヒットカウント値が1つインクリメントされ、

前記FSA方式キャッシュ (TA) と前記FSA方式以外のキャッシュ (TB, TC) の両方がミスヒットした場合には、前記FSA方式キャッシュ (TA) の全キャッシュラインの前記キャッシュヒットカウント値が一斉に1つデクリメントされることを特徴とする請求項10記載のキャッシュ処理方法。

【請求項12】 前記キャッシュヒットカウント管理工

10

20

30

40

50

程において、

前記FSA方式キャッシュ(TA)のあるキャッシュラインがキャッシュヒットした場合には、そのキャッシュラインの前記キャッシュヒットカウント値が1つインクリメントされ、

前記FSA方式キャッシュ(TA)がミスヒットした場合には、前記FSA方式キャッシュ(TA)の全キャッシュラインの前記キャッシュヒットカウント値が一斉に1つデクリメントされることを特徴とする請求項10記載のキャッシュ処理方法。

【請求項13】 前記キャッシュヒットカウント管理工程において、

前記FSA方式キャッシュ(TA)のあるキャッシュラインがキャッシュヒットした場合には、そのキャッシュラインの前記キャッシュヒットカウント値が1つインクリメントされることを特徴とする請求項10記載のキャッシュ処理方法。

【請求項14】 前記FSA方式以外のキャッシュ(TB, TC)として、ダイレクトマッピング方式キャッシュ(TB)が用いられることを特徴とする請求項10から13のうちのいずれか1項に記載のキャッシュ処理方法。

【請求項15】 前記FSA方式以外のキャッシュ(TB, TC)として、Nウェイセットアソシアティブキャッシュ(TC)(N=2, 4, 8, ...)が用いられることを特徴とする請求項10から13のうちのいずれか1項に記載のキャッシュ処理方法。

【請求項16】 前記FSA方式キャッシュ(TA)の各キャッシュラインに生じた最新のキャッシュヒットの日時に関するデータを、前記各キャッシュラインに対応して設けられたキャッシュヒット日時格納手段に格納するキャッシュヒット日時格納工程をさらに備え、前記第2のデータ転送工程においては、前記キャッシュヒット日時格納手段に格納された前記データに基づいて前記FSA方式キャッシュ(TA)の前記最小のキャッシュヒットカウント値を持つ前記キャッシュラインの中から最近のキャッシュヒットが最も古いキャッシュラインが選択され、その選択されたキャッシュラインが前記FSA方式以外のキャッシュ(TB, TC)から前記FSA方式キャッシュ(TA)への前記データ転送の転送先として指定されることを特徴とする請求項10から15のうちのいずれか1項に記載のキャッシュ処理方法。

【請求項17】 各インデックスに対応するN本のキャッシュラインの中で最近のキャッシュヒットが最も古いキャッシュラインを示すためのLRU(Least Recently Used)値を各インデックスに対応して設けられたLRU格納手段に格納するLRU格納工程をさらに備え、

前記第1のデータ転送工程においては、前記Nウェイセットアソシアティブキャッシュ(TC)のN本のミスヒ

10

20

30

40

50

ットしたキャッシュラインの中から前記LRU値により指定されたキャッシュラインが選択され、その選択されたキャッシュラインに格納されていたデータが前記FSA方式キャッシュ(TA)の前記非活性なキャッシュラインのうちの一つに転送され、前記両方のキャッシュのミスヒットにより前記外部メモリ(20, 21)からフェッチされたデータは前記FSA方式キャッシュ(TA)へのデータ転送元となった前記Nウェイセットアソシアティブキャッシュ(TC)の前記選択されたキャッシュラインに書き込まれ、

前記第2のデータ転送工程においては、前記Nウェイセットアソシアティブキャッシュ(TC)のN本のミスヒットしたキャッシュラインの中から前記LRU値により指定されたキャッシュラインが選択され、その選択されたキャッシュラインに格納されていたデータが前記FSA方式キャッシュ(TA)の前記最小のキャッシュヒットカウント値を持つ前記キャッシュラインのうちの一つに転送され、前記両方のキャッシュのミスヒットにより前記外部メモリ(20, 21)からフェッチされたデータは前記FSA方式キャッシュ(TA)へのデータ転送元となった前記Nウェイセットアソシアティブキャッシュ(TC)の前記選択されたキャッシュラインに書き込まれ、

前記第3のデータ転送工程においては、前記両方のキャッシュのミスヒットにより前記外部メモリ(20, 21)からフェッチされたデータは前記Nウェイセットアソシアティブキャッシュ(TC)の前記インデックスに対応する前記非活性なキャッシュラインのうちの一つのキャッシュラインに書き込まれることを特徴とする請求項15記載のキャッシュ処理方法。

【請求項18】 前記FSA方式キャッシュ(TA)と前記FSA方式以外のキャッシュ(TB, TC)の両方がミスヒットした場合に、前記FSA方式キャッシュ

(TA)と前記FSA方式以外のキャッシュ(TB, TC)からなる一次キャッシュ(15)に付加する形で設けられた二次キャッシュ(20)へのアクセスが行われる二次キャッシュアクセス工程をさらに備えることを特徴とする請求項10から17のうちのいずれか1項に記載のキャッシュ処理方法。

【請求項19】 キャッシュヒット率の高いキャッシュラインの保持が可能なFSA(フルセットアソシアティブ)方式キャッシュ(TA)と前記FSA方式キャッシュ(TA)と同時にキャッシュヒット判定のためのタグ比較を行うFSA方式以外のキャッシュ(TB, TC)との組み合わせを用いてCPU(Central Processing Unit)(50)、MPU(MicroProcessor Unit)等のデータ処理ユニット(50)のメインメモリ(21)等の外部メモリ(20, 21)へのアクセス回数を削減するためのキャッシュ処理をコンピュータに実行させるためのプログ

ラムを格納した機械読み取り可能な記録媒体であり、前記キャッシュ処理は、

前記FSA方式キャッシュ(TA)の各キャッシュラインに起きたキャッシュヒットがカウントされ、前記FSA方式キャッシュ(TA)の各キャッシュラインに対応して設けられたキャッシュヒットカウント格納手段(AC)に格納された各キャッシュライン毎のキャッシュヒットカウント値が管理し更新されるキャッシュヒットカウント管理工程と、

前記FSA方式キャッシュ(TA)がキャッシュヒットした場合に、前記FSA方式キャッシュ(TA)のそのキャッシュヒットしたキャッシュラインに格納されていたデータが、前記データ処理ユニット(50)によりキャッシュデータとして読み込まれる第1のキャッシュデータ読み込み工程と、

前記FSA方式以外のキャッシュ(TB, TC)がキャッシュヒットした場合に、前記FSA方式以外のキャッシュ(TB, TC)の、入力アドレスデータから抽出されたインデックスに対応するそのキャッシュヒットしたキャッシュラインに格納されていたデータが、前記データ処理ユニット(50)によりキャッシュデータとして読み込まれる第2のキャッシュデータ読み込み工程と、前記FSA方式以外のキャッシュ(TB, TC)に前記インデックスに対応する非活性なキャッシュラインが存在せず前記FSA方式キャッシュ(TA)に1以上の非活性なキャッシュラインが存在する時に前記FSA方式キャッシュ(TA)と前記FSA方式以外のキャッシュ(TB, TC)の両方がミスヒットした場合に、前記FSA方式以外のキャッシュ(TB, TC)のミスヒットした前記インデックスに対応するあるキャッシュラインに格納されていたデータが前記FSA方式キャッシュ

(TA)の前記非活性なキャッシュラインのうちの一つに転送され、前記両方のキャッシュのミスヒットにより前記外部メモリ(20, 21)からフェッチされたデータが前記FSA方式キャッシュ(TA)へのデータ転送元となった前記FSA方式以外のキャッシュ(TB, TC)の前記キャッシュラインに書き込まれる第1のデータ転送工程と、

前記FSA方式以外のキャッシュ(TB, TC)に前記インデックスに対応する非活性なキャッシュラインが存在せず前記FSA方式キャッシュ(TA)が活性なキャッシュラインで一杯な時に前記FSA方式キャッシュ

(TA)と前記FSA方式以外のキャッシュ(TB, TC)の両方がミスヒットした場合に、前記FSA方式以外のキャッシュ(TB, TC)のミスヒットした前記インデックスに対応するあるキャッシュラインに格納されていたデータが前記FSA方式キャッシュ(TA)の最小のキャッシュヒットカウント値を持つキャッシュラインのうちの一つに転送され、前記両方のキャッシュのミスヒットにより前記外部メモリ(20, 21)からフェ

ッチされたデータが前記FSA方式キャッシュ(TA)へのデータ転送元となった前記FSA方式以外のキャッシュ(TB, TC)の前記キャッシュラインに書き込まれる第2のデータ転送工程と、

前記FSA方式以外のキャッシュ(TB, TC)に前記インデックスに対応する非活性なキャッシュラインが存在する時に前記FSA方式キャッシュ(TA)と前記FSA方式以外のキャッシュ(TB, TC)の両方がミスヒットした場合に、前記両方のキャッシュのミスヒットにより前記外部メモリ(20, 21)からフェッチされたデータが前記FSA方式以外のキャッシュ(TB, TC)の前記インデックスに対応する非活性なキャッシュラインに書き込まれる第3のデータ転送工程とを備えることを特徴とする機械読み取り可能な記録媒体。

【請求項20】 前記キャッシュヒットカウント管理工程において、

前記FSA方式キャッシュ(TA)のあるキャッシュラインがキャッシュヒットした場合には、そのキャッシュラインの前記キャッシュヒットカウント値が1つインクリメントされ、

前記FSA方式キャッシュ(TA)と前記FSA方式以外のキャッシュ(TB, TC)の両方がミスヒットした場合には、前記FSA方式キャッシュ(TA)の全キャッシュラインの前記キャッシュヒットカウント値が一斉に1つデクリメントされることを特徴とする請求項19記載の機械読み取り可能な記録媒体。

【請求項21】 前記キャッシュヒットカウント管理工程において、

前記FSA方式キャッシュ(TA)のあるキャッシュラインがキャッシュヒットした場合には、そのキャッシュラインの前記キャッシュヒットカウント値が1つインクリメントされ、

前記FSA方式キャッシュ(TA)がミスヒットした場合には、前記FSA方式キャッシュ(TA)の全キャッシュラインの前記キャッシュヒットカウント値が一斉に1つデクリメントされることを特徴とする請求項19記載の機械読み取り可能な記録媒体。

【請求項22】 前記キャッシュヒットカウント管理工程において、

前記FSA方式キャッシュ(TA)のあるキャッシュラインがキャッシュヒットした場合には、そのキャッシュラインの前記キャッシュヒットカウント値が1つインクリメントされることを特徴とする請求項19記載の機械読み取り可能な記録媒体。

【請求項23】 前記FSA方式以外のキャッシュ(TB, TC)として、ダイレクトマッピング方式キャッシュ(TB)が用いられることを特徴とする請求項19から22のうちのいずれか1項に記載の機械読み取り可能な記録媒体。

【請求項24】 前記FSA方式以外のキャッシュ(T

10

20

30

40

50



B, TC) として、Nウェイセットアソシアティブキャッシュ (TC) (N=2, 4, 8, ...) が用いられることを特徴とする請求項 19 から 22 のうちのいずれか 1 項に記載の機械読み取り可能な記録媒体。

【請求項 25】 前記キャッシュ処理は、前記 FSA 方式キャッシュ (TA) の各キャッシュラインに生じた最新のキャッシュヒットの日時に関するデータを前記各キャッシュラインに対応して設けられたキャッシュヒット日時格納手段に格納するキャッシュヒット日時格納工程をさらに備え、

前記第 2 のデータ転送工程においては、前記キャッシュヒット日時格納手段に格納された前記データに基づいて前記 FSA 方式キャッシュ (TA) の前記最小のキャッシュヒットカウント値を持つ前記キャッシュラインの中から最近のキャッシュヒットが最も古いキャッシュラインが選択され、その選択されたキャッシュラインが前記 FSA 方式以外のキャッシュ (TB, TC) から前記 FSA 方式キャッシュ (TA) への前記データ転送の転送先として指定されることを特徴とする請求項 19 から 24 のうちのいずれか 1 項に記載の機械読み取り可能な記録媒体。

【請求項 26】 前記キャッシュ処理は、各インデックスに対応する N 本のキャッシュラインの中で最近のキャッシュヒットが最も古いキャッシュラインを示すための LRU (Least Recently Used) 値を各インデックスに対応して設けられた LRU 格納手段に格納する LRU 格納工程をさらに備え、

前記第 1 のデータ転送工程においては、前記 Nウェイセットアソシアティブキャッシュ (TC) の N 本のミスヒットしたキャッシュラインの中から前記 LRU LRU 値により指定されたキャッシュラインが選択され、その選択されたキャッシュラインに格納されていたデータが前記 FSA 方式キャッシュ (TA) の前記非活性なキャッシュラインのうちの一つに転送され、前記両方のキャッシュのミスヒットにより前記外部メモリ (20, 21) からフェッチされたデータは前記 FSA 方式キャッシュ (TA) へのデータ転送元となった前記 Nウェイセットアソシアティブキャッシュ (TC) の前記選択されたキャッシュラインに書き込まれ、

前記第 2 のデータ転送工程においては、前記 Nウェイセットアソシアティブキャッシュ (TC) の N 本のミスヒットしたキャッシュラインの中から前記 LRU 値により指定されたキャッシュラインが選択され、その選択されたキャッシュラインに格納されていたデータが前記 FSA 方式キャッシュ (TA) の前記最小のキャッシュヒットカウント値を持つ前記キャッシュラインのうちの一つに転送され、前記両方のキャッシュのミスヒットにより前記外部メモリ (20, 21) からフェッチされたデータは前記 FSA 方式キャッシュ (TA) へのデータ転送元となった前記 Nウェイセットアソシアティブキャッシュ

ユ (TC) の前記選択されたキャッシュラインに書き込まれ、

前記第 3 のデータ転送工程においては、前記両方のキャッシュのミスヒットにより前記外部メモリ (20, 21) からフェッチされたデータは前記 Nウェイセットアソシアティブキャッシュ (TC) の前記インデックスに対応する前記非活性なキャッシュラインのうちの一つのキャッシュラインに書き込まれることを特徴とする請求項 24 記載の機械読み取り可能な記録媒体。

10 【請求項 27】 前記キャッシュ処理は、前記 FSA 方式キャッシュ (TA) と前記 FSA 方式以外のキャッシュ (TB, TC) の両方がミスヒットした場合に前記 FSA 方式キャッシュ (TA) と前記 FSA 方式以外のキャッシュ (TB, TC) からなる一次キャッシュ (15) に付加する形で設けられた二次キャッシュ (20) へのアクセスが行われる二次キャッシュアクセス工程をさらに備えることを特徴とする請求項 19 から 26 のうちのいずれか 1 項に記載の機械読み取り可能な記録媒体。

20 【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、MPU (Micro Processor Unit)、CPU (Central Processing Unit) 等のデータ処理ユニットに適用され、メインメモリ等の外部メモリへのアクセス時間を短縮するためのキャッシュシステムおよびキャッシュ処理方法に関するものであり、特に、FSA (フルセットアソシアティブ) 方式以外のキャッシュ (ダイレクトマッピング方式キャッシュ、2ウェイアソシアティブ方式キャッシュ、4ウェイアソシアティブ方式キャッシュ、8ウェイアソシアティブ方式キャッシュ等) に、使用頻度の高いキャッシュラインを保持させることが可能な FSA 方式キャッシュを付加することによりキャッシュヒット率の向上を図るキャッシュシステムおよびキャッシュ処理方法に関するものである。

【0002】

【従来の技術】 キャッシュ (キャッシュメモリともいう) は、高速でデータ処理を実行する MPU、CPU 等がデータ処理速度の低いメインメモリ等の外部メモリへアクセスする際のアクセス時間を短縮するために広く用いられている。特に、MPU、CPU 等のプログラム実行速度を高め、MPU、CPU 等を含むシステムのスループットを向上させるために一次キャッシュと二次キャッシュとからなる階層化キャッシュが多用されている。キャッシュメモリは一般に、複数のタグと各々のタグに対応したデータとを保持するキャッシュテーブルを備えており、入力データから分離されたタグがキャッシュテーブルに格納されたタグと比較される。タグが一致した際には、一致したタグに対応したデータがキャッシュテーブルから選択され、MPU、CPU 等に供給される。

このような処理により、データ処理速度の低いメインメモリ等の外部メモリへのアクセス回数が抑えられ、MPU、CPU等のデータ処理ユニットの高速データ処理が実現される。

【0003】このようなキャッシュとしては、FSA方式キャッシュとダイレクトマッピング方式キャッシュがよく知られている。ダイレクトマッピング方式キャッシュは、小回路規模で高速アクセスを実現することができるが、そのキャッシュヒット率が悪化しやすい。一方、FSA方式キャッシュは消費電力が大きくかつ回路規模が複雑であるものの、ヒット率の高いキャッシュラインを保持することが可能である。なお、ダイレクトマッピング方式キャッシュと類似する機能を備えたFSA方式以外のキャッシュとして、2ウェイアソシアティブ方式キャッシュ、4ウェイアソシアティブ方式キャッシュ、8ウェイアソシアティブ方式キャッシュも周知である。

【0004】図10は、典型的な従来のダイレクトマッピング方式キャッシュを示すブロック図である。図10のダイレクトマッピング方式キャッシュは、複数のタグとこれらのタグに対応するデータとを格納するキャッシュテーブル801と、コンパレータ802と、ANDゲート803と、データセクタ804とを備えている。キャッシュテーブル801の各キャッシュラインには、そのキャッシュラインが活性か非活性かを示すための活性化ビット(valid bit)が設けられている。活性化ビット「1」はそのキャッシュラインが活性であることを示し、活性化ビット「0」はそのキャッシュラインが非活性であることを示す。図10の上部には、入力アドレスデータの例として「00000040」(16進)が示されている。この入力アドレスデータは、タグ、インデックス、およびオフセットを含んでおり、例えば入力アドレスデータ「00000040」の場合には、タグは「00000」(例えば16進数の入力アドレスデータの最初の5桁)、インデックスは「04」(例えば16進数の入力アドレスデータの次の2桁)、オフセットは「0」(例えば16進数の入力アドレスデータの最後の1桁)となる。

【0005】図11はCPUにより実現されるプログラムの例を示す模式図である。図11のプログラムは、実行されるべき複数の命令(1)、(2)、・・・を含んでいる。メインメモリには予め多数の命令がそれぞれ対応するアドレスに格納されており、図11のプログラムがCPUによって実行される場合には、CPUはまずこのプログラムの最初の命令(1)に対応する入力アドレスデータ(プログラムカウンタから与えられる)を参照する。最初の命令(1)に対応する入力アドレスデータ「00000040」は、この命令(1)がメインメモリのアドレス「00000040」に予め格納されていることを示す。この最初の命令(1)に対して、図10のダイレクトマッピング方式キャッシュのコンパレータ

802が、入力アドレスデータ「00000040」から抽出されたタグ「00000」が図10のキャッシュテーブル801のインデックス「04」に対応するキャッシュラインに格納されていたタグに一致するかを判定する。一致した場合には、その一致したタグに対応するデータがダイレクトマッピング方式キャッシュのキャッシュテーブル801から読み出され、CPU(命令デコーダ)に送られる。一致しなかった場合は、CPUはメインメモリにアクセスし、メインメモリのアドレス「00000040」からこの命令(1)をフェッチする。この不一致の場合、インデックス「04」に対応するキャッシュラインは書き換えられる。すなわち、インデックス「04」に対応するキャッシュラインのデータはメインメモリからフェッチされたデータに書き換えられ、そのキャッシュラインのタグは入力アドレスデータ「00000040」に対応するタグ「00000」に書き換えられる。その後、続く命令(2)、(3)・・・に対しても同様の処理が行われる。このようなダイレクトマッピング方式キャッシュの使用により、CPUの長いアクセス時間を要するメインメモリへのアクセスの回数が減られ、CPUによる高速プログラム実行(キャッシュ処理)が実現される。

【0006】図12(a)および(b)は、CPUが図11のプログラムを実行する際の図10のダイレクトマッピング方式キャッシュのキャッシュテーブル801の状態変化の例を示す模式図である。図12(a)は命令(1)実行直後の状態を示し、図12(b)は命令(5)実行直後の状態を示す。12(a)を参照すると、命令(1)が実行された直後の時点では、キャッシュテーブル801のインデックス「04」に対応するキャッシュラインには、入力アドレスデータ「00000040」に対応するタグ「00000」とこのタグ「00000」に対応するデータとが格納されている。図12(b)を参照すると、命令(5)が実行された直後の時点では、キャッシュテーブル801のインデックス「04」に対応する同じキャッシュラインには、入力アドレスデータ「00001040」に対応するタグ「00001」とこのタグ「00001」に対応するデータとが格納されている。

【0007】図13(a)および(b)は、このダイレクトマッピング方式キャッシュを使用するCPUが図11のプログラムを2回実行する場合のアクセス時間を示す模式図であり、図13(a)は1回目のプログラム実行の場合、図13(b)は2回目のプログラム実行の場合を示している。なお、以下のプログラム実行時間に関する説明は、ダイレクトマッピング方式キャッシュの各キャッシュラインのデータ格納領域の長さが4ワード(16バイト)であり図11のプログラムの各命令の長さが1ワード(4バイト)である(すなわち、ダイレクトマッピング方式キャッシュの各キャッシュラインのデ

ータ格納領域に4個の命令が格納される)という仮定の下に行う。データ(命令)をメインメモリからフェッチするのに必要なアクセス時間は、最初の1ワードに100ns、続く3ワードの各々にそれぞれ30nsと仮定する。従って、4ワード分のデータ(4個の命令)をメインメモリからフェッチしてこれらのデータをダイレクトマッピング方式キャッシュの1つのキャッシュラインに格納するために必要なアクセス時間は、 $100+30+30+30=190\text{ns}$ となる。CPUがキャッシュラインに格納された直後の命令を読み出して実行するのに10nsかかるかと仮定すれば、4ワード分のデータ(4個の命令)をメインメモリからフェッチしてこれらのデータをダイレクトマッピング方式キャッシュに格納して格納された最初の命令(1)を実行するのに要するアクセス時間は $190+10=200\text{ns}$ となる。

【0008】初期化において、ダイレクトマッピング方式キャッシュのすべてのキャッシュラインが非活性化される。すなわち、全キャッシュラインの活性化ビットが「0」にリセットされる。この初期化の直後には活性化ビットVがすべて「0」であるため、ダイレクトマッピング方式キャッシュ中には実質的にはデータが存在しないことになる。このためCPUは、必要なデータ(最初のインデックス「04」に対応する命令(1)～

(4))をメインメモリからフェッチし、フェッチした命令(1)～(4)をダイレクトマッピング方式キャッシュのインデックス「04」に対応するキャッシュラインに書き込む。このデータ書き込みにより、このインデックス「04」に対応するキャッシュラインは活性化される。すなわち、このキャッシュラインの活性化ビットVが「1」にセットされる(図10参照)。その後、CPUはこのキャッシュラインから命令(1)を読み出して実行する。図13(a)に示される最初のプログラム実行においては、スタートアップルーチンの後にCPUが最初の命令(1)を実行するのに $100+30+30+30+10=200\text{ns}$ が必要であり、続く命令

(2)、(3)、(4)の実行には各々10nsが必要である(230ns)。その後CPUは、アドレス「00001030」から「0000103C」に対応する4個の命令を同様に実行する( $200+10+10+10=230\text{ns}$ )。

【0009】その後、前述のインデックス「04」に対応する命令(5)をCPUが実行しようとする時、ダイレクトマッピング方式キャッシュのインデックス「04」に対応するキャッシュラインには、既に命令(1)とこの命令(1)に対応するタグ「00000」が格納されているため、このインデックス「04」に対応するキャッシュラインにミスヒット(「00001」≠「00000」)が起こる。このため、CPUは命令(5)をメインメモリからフェッチし、この命令(5)と対応するタグ「00001」とをインデックス「04」に

応するキャッシュラインに格納する。このミスヒットにより、命令(5)の実行も $100+30+30+30+10=200\text{ns}$ かかる。続く3個の命令の実行には同様に各々10nsかかる。従って、1回目のプログラム実行における総プログラム実行時間は $230\times 3=690\text{ns}$ となる。

【0010】図13(a)の1回目のプログラム実行が終了した時点で、インデックス「04」に対応するキャッシュラインは、図12(b)に示すように、命令(5)に対応するタグ「00001」を格納している。このため、2回目のプログラム実行において前述のインデックス「04」に対応する命令(1)をCPUが実行しようとする時、このインデックス「04」に対応するキャッシュラインに再度ミスヒット(「00000」≠「00001」)が起こる。このためCPUはこの命令(1)を再度メインメモリからフェッチすることとなり、2回目のプログラム実行における命令(1)の実行にまた200nsかかることとなる。続く3個の命令の実行には同様に各々10nsかかる(230ns)。その後CPUは、アドレス「00001030」から「0000103C」に対応する4個の命令を、ダイレクトマッピング方式キャッシュのインデックス「03」に対応するキャッシュラインに既に格納されたキャッシュデータを使用して実行する( $10+10+10+10=40\text{ns}$ )。その後、2回目のプログラム実行において前述のインデックス「04」に対応する命令(5)をCPUが実行しようとする時、このインデックス「04」に対応するキャッシュラインに再度同様なミスヒット

(「00001」≠「00000」)が起こる(200ns)。続く3個の命令の実行には同様に各々10nsかかる(230ns)。従って、2回目のプログラム実行におけるプログラム実行時間は $230+40+230=500\text{ns}$ となる。よって、図11のプログラムを2回実行するために必要な総プログラム実行時間(アクセス時間)は $690+500=1190\text{ns}$ となる。

【0011】上述のように、ダイレクトマッピング方式キャッシュは小さな回路規模で高速アクセスを実現できる長所を持つ反面、キャッシュミス(ミスヒット)が容易にかつ繰り返して発生してしまう欠点を持つ。ミスヒットによるメインメモリへのアクセスは例えば200nsという長時間を要し、このためにCPUのプログラム実行時間(プログラム実行のためのアクセス時間)が長くなってしまふという課題があった。

【0012】キャッシュヒット率を向上させ、メモリアクセス時間およびプログラム実行時間を低減するため、一次キャッシュと二次キャッシュとからなる階層化キャッシュが広く用いられるようになった。例えば、特開昭61-241853号公報に開示された「キャッシュメモリ制御方式」の従来例では、一次キャッシュがCPUとメインメモリの間に設けられ、二次キャッシュが一次

10

20

30

40

50

キャッシュとメインメモリの間に設けられる。一次キャッシュがキャッシュヒットした場合、一次キャッシュに格納されていた必要なデータがキャッシュデータとしてCPUに与えられる。一次キャッシュがミスヒットし、二次キャッシュがキャッシュヒットした場合、二次キャッシュに格納されていた必要なデータが一次キャッシュに転送され、その後CPUに与えられる。この場合、二次キャッシュから転送された必要なデータを格納できる非活性な（使用されていない）キャッシュラインが一次キャッシュになれば、一次キャッシュの1つのキャッシュラインに格納されていたデータが一次キャッシュから削除され、二次キャッシュの1つのキャッシュラインに転送される。この場合、一次キャッシュから削除されたデータを格納できる非活性な（使用されていない）キャッシュラインが二次キャッシュになれば、その一次キャッシュから削除されたデータを格納できるように、二次キャッシュの1つのキャッシュラインに格納されていたデータが二次キャッシュから削除される。一次キャッシュと二次キャッシュの両方がミスヒットした場合、必要なデータはメインメモリからフェッチされ、一次キャッシュに書き込まれる。通常の階層化キャッシュではフェッチされたデータは一次キャッシュと二次キャッシュの両方に書き込まれるが、この従来例では二次キャッシュには書き込まれず一次キャッシュにのみ書き込まれる。このような動作により、二次キャッシュのキャッシュヒット率の改善が図られている。

【0013】また、特開平5-73415号公報に開示された「階層化キャッシュ方式」の従来例では、CPUが、必要なデータが一次キャッシュか二次キャッシュに存在するか否かを判別する。一次キャッシュがミスヒットし、二次キャッシュがヒットした場合、二次キャッシュでヒットしたデータは一次キャッシュに転送される。この転送において、一次キャッシュから排除されることとなるデータは二次キャッシュに転送される。

【0014】また、特開平6-012323号公報に開示された「キャッシュメモリシステム」の従来例では、ダイレクトマッピング方式キャッシュが一次キャッシュに、FSA方式キャッシュが二次キャッシュに用いられ、キャッシュヒットの判定のためのタグ比較が一次キャッシュと二次キャッシュとで同時に行われる。メインメモリからフェッチされたデータは、従来の階層化キャッシュの技術と同様にして一次キャッシュと二次キャッシュとの両方に書き込まれる。この従来例には二次キャッシュ（FSA方式キャッシュ）の各キャッシュラインへのアクセス（キャッシュヒット）の回数をカウントするカウンタが提案されており、1つのキャッシュラインのカウンタのカウント値が所定数を越えた場合には、そのキャッシュラインが後のアクセスのために一次キャッシュ（ダイレクトマッピング方式キャッシュ）にロードされる。

【0015】また、特開平6-250926号公報に開示された「複数階層のキャッシュメモリを備えたデータ処理システム」の従来例では、一次キャッシュがミスヒットした場合には二次キャッシュへのアクセスが行われる。二次キャッシュがキャッシュヒットした場合、二次キャッシュに格納されていた必要なデータが一次キャッシュに転送され（二次キャッシュにおいては非活性化される）、CPUに与えられる。二次キャッシュでもミスヒットした場合、メインメモリへのアクセスが行われ、必要なデータがメインメモリからフェッチされる。この従来例のある実施の形態では、一次キャッシュに十分なメモリ空間がある場合にはメインメモリからフェッチされたデータは一次キャッシュにのみ登録され（二次キャッシュには登録されない）、一次キャッシュに十分なメモリ空間がない場合にはメインメモリからフェッチされたデータは二次キャッシュにのみ登録される（一次キャッシュには登録されない）。他の実施の形態では、メインメモリからフェッチされたデータは一次キャッシュのみに直接登録される（二次キャッシュには登録されない）。このメインメモリから一次キャッシュへの直接データ登録において一次キャッシュに登録できないデータが生じた場合、そのデータは一次キャッシュから転送されて二次キャッシュに登録される。

#### 【0016】

【発明が解決しようとする課題】前述のように、従来のダイレクトマッピング方式キャッシュにおいては、キャッシュヒット率の高いキャッシュラインの保持を行うことは不可能である。ダイレクトマッピング方式キャッシュメモリは1つのインデックスに対して1つのキャッシュラインしか持てないためミスヒットが頻繁にかつ反復的に発生し、そのためメインメモリへのアクセス回数が多くなってしまうという課題があった。

【0017】また、一次キャッシュと二次キャッシュを有した階層化キャッシュを用いることによりキャッシュヒット率を改善することが可能となったが、二次キャッシュへのアクセスは一般に余分なアクセス時間を必要とし、CPU等のデータ処理ユニットのプログラム実行時間が長くなってしまうという課題があった。タグ比較をダイレクトマッピング方式キャッシュによりなる一次キャッシュとFSA方式キャッシュによりなる二次キャッシュとの両方で同時に行う階層化キャッシュの従来例も存在したが、FSA方式キャッシュにおけるヒット率の高いキャッシュラインの保持に関して改善の余地がまだ多く残るものであった。また、同一のデータが一次キャッシュ（ダイレクトマッピング方式キャッシュ）と二次キャッシュ（FSA方式キャッシュ）との両方に格納される場合が存在し、そのためにキャッシュメモリのメモリ空間の使用効率が低下し、キャッシュメモリに格納可能なキャッシュデータの量が減少してしまい、キャッシュヒット率を最大まで向上させることができないという

課題があった。

【0018】本発明は、このような従来の技術における課題を解決するものであり、FSA方式キャッシュとダイレクトマッピング方式キャッシュ等のFSA方式以外のキャッシュとを組み合わせ、ヒット率の高いキャッシュラインの保持性能とキャッシュメモリ空間の使用効率を向上することにより、高速データアクセスと高キャッシュヒット率の両方を高レベルで実現し、これによりCPU、MPU等のデータ処理ユニットの、メインメモリ等の外部メモリへのアクセス回数とデータアクセス時間とを最小化することが可能なキャッシュシステムおよびキャッシュ処理方法を提供することを目的とする。

【0019】

【課題を解決するための手段】請求項1記載のキャッシュシステムは、CPU (Central Processing Unit)、MPU (MicroProcessor Unit) 等のデータ処理ユニットの、メインメモリ等の外部メモリへのアクセス回数を削減するためのキャッシュシステムにおいて、キャッシュヒット率の高いキャッシュラインの保持が可能なFSA (フルセットアソシアティブ) 方式キャッシュであり入力アドレスデータから抽出された第1のタグが前記FSA方式キャッシュの活性なキャッシュラインに格納されていた第1のタグの一つと一致した場合にその一致した第1のタグに対応する活性なキャッシュラインに格納されていたデータが前記データ処理ユニットによりキャッシュデータとして読み込まれるFSA方式キャッシュと、前記キャッシュシステムにおいて前記FSA方式キャッシュと組み合わせられる形で設けられキャッシュヒット判定のためのタグ比較を前記FSA方式キャッシュと同時に行うFSA方式以外のキャッシュであり前記入力アドレスデータから抽出された第2のタグが前記FSA方式以外のキャッシュの前記入力アドレスデータから抽出されたインデックスに対応する活性なキャッシュラインに格納されていた第2のタグに一致した場合にその一致した第2のタグに対応する活性なキャッシュラインに格納されていたデータが前記データ処理ユニットによりキャッシュデータとして読み込まれるFSA方式以外のキャッシュと、前記FSA方式キャッシュの各キャッシュラインに対応して設けられ前記キャッシュラインに生じたキャッシュヒットの回数に関するキャッシュヒットカウント値を格納するキャッシュヒットカウント格納手段と、前記FSA方式キャッシュの各キャッシュラインのキャッシュヒットをカウントし前記キャッシュヒットカウント格納手段に格納される前記キャッシュヒットカウント値を管理し更新するキャッシュヒットカウント管理手段とを備えるようにし、前記FSA方式以外のキャッシュに前記インデックスに対応する非活性なキャッシュラインが存在せず前記FSA方式キャッシュに1以上の非活性なキャッシュラインが存在する時に前記FSA方式キャ

ッシュと前記FSA方式以外のキャッシュの両方がミスヒットした場合は、前記FSA方式以外のキャッシュのミスヒットした前記インデックスに対応するあるキャッシュラインに格納されていたデータが前記FSA方式キャッシュの前記非活性なキャッシュラインのうちの一つに転送され、前記両方のキャッシュのミスヒットにより前記外部メモリからフェッチされたデータは前記FSA方式キャッシュへのデータ転送元となった前記FSA方式以外のキャッシュの前記キャッシュラインに書き込まれるようにし、前記FSA方式以外のキャッシュに前記インデックスに対応する非活性なキャッシュラインが存在せず前記FSA方式キャッシュが活性なキャッシュラインで一杯な時に前記FSA方式キャッシュと前記FSA方式以外のキャッシュの両方がミスヒットした場合は、前記FSA方式以外のキャッシュのミスヒットした前記インデックスに対応するあるキャッシュラインに格納されていたデータが前記FSA方式キャッシュの最小のキャッシュヒットカウント値を持つキャッシュラインのうちの一つに転送され、前記両方のキャッシュのミスヒットにより前記外部メモリからフェッチされたデータは前記FSA方式キャッシュへのデータ転送元となった前記FSA方式以外のキャッシュの前記キャッシュラインに書き込まれるようにし、前記FSA方式以外のキャッシュに前記インデックスに対応する非活性なキャッシュラインが存在する時に前記FSA方式キャッシュと前記FSA方式以外のキャッシュの両方がミスヒットした場合は、前記両方のキャッシュのミスヒットにより前記外部メモリからフェッチされたデータは前記FSA方式以外のキャッシュの前記インデックスに対応する非活性なキャッシュラインに書き込まれるようにしたものである。

【0020】請求項2記載のキャッシュシステムは、請求項1記載のキャッシュシステムにおいて、前記FSA方式キャッシュのあるキャッシュラインがキャッシュヒットした場合には、前記キャッシュヒットカウント管理手段はそのキャッシュラインの前記キャッシュヒットカウント値を1つインクリメントし、前記FSA方式キャッシュと前記FSA方式以外のキャッシュの両方がミスヒットした場合には、前記キャッシュヒットカウント管理手段は前記FSA方式キャッシュの全キャッシュラインの前記キャッシュヒットカウント値を一斉に1つデクリメントするようにしたものである。

【0021】請求項3記載のキャッシュシステムは、請求項1記載のキャッシュシステムにおいて、前記FSA方式キャッシュのあるキャッシュラインがキャッシュヒットした場合には、前記キャッシュヒットカウント管理手段はそのキャッシュラインの前記キャッシュヒットカウント値を1つインクリメントし、前記FSA方式キャッシュがミスヒットした場合には、前記キャッシュヒットカウント管理手段は前記FSA方式キャッシュの全キ

キャッシュラインの前記キャッシュヒットカウント値を一斉に1つデクリメントするようにしたものである。

【0022】請求項4記載のキャッシュシステムは、請求項1記載のキャッシュシステムにおいて、前記FSA方式キャッシュのあるキャッシュラインがキャッシュヒットした場合には、前記キャッシュヒットカウント管理手段はそのキャッシュラインの前記キャッシュヒットカウント値を1つインクリメントするようにしたものである。

【0023】請求項5記載のキャッシュシステムは、請求項1から4のうちのいずれか1項に記載のキャッシュシステムにおいて、前記FSA方式以外のキャッシュとして、ダイレクトマッピング方式キャッシュを用いるものである。

【0024】請求項6記載のキャッシュシステムは、請求項1から4のうちのいずれか1項に記載のキャッシュシステムにおいて、前記FSA方式以外のキャッシュとして、Nウェイセットアソシアティブキャッシュ(N=2, 4, 8, ...)が用いるものである。

【0025】請求項7記載のキャッシュシステムは、請求項1から6のうちのいずれか1項に記載のキャッシュシステムにおいて、前記FSA方式キャッシュの各キャッシュラインに対応して設けられ、前記キャッシュラインに生じた最新のキャッシュヒットの日時に関するデータを格納するキャッシュヒット日時格納手段をさらに備えるようにし、前記FSA方式以外のキャッシュに前記インデックスに対応する非活性なキャッシュラインが存在せず前記FSA方式キャッシュが活性なキャッシュラインで一杯な時に前記FSA方式キャッシュと前記FSA方式以外のキャッシュの両方がミスヒットした前記場合には、前記キャッシュヒット日時格納手段に格納された前記データに基づいて前記FSA方式キャッシュの前記最小のキャッシュヒットカウント値を持つ前記キャッシュラインの中から最近のキャッシュヒットが最も古いキャッシュラインが選択され、その選択されたキャッシュラインが前記FSA方式以外のキャッシュから前記FSA方式キャッシュへの前記データ転送の転送先として指定されるようにしたものである。

【0026】請求項8記載のキャッシュシステムは、請求項6記載のキャッシュシステムにおいて、各インデックスに対応して設けられ、そのインデックスに対応するN本のキャッシュラインの中で最近のキャッシュヒットが最も古いキャッシュラインを示すためのLRU (Least Recently Used) 値を格納するLRU格納手段をさらに備えるようにし、前記Nウェイセットアソシアティブキャッシュに前記インデックスに対応する非活性なキャッシュラインが存在せず前記FSA方式キャッシュに1以上の非活性なキャッシュラインが存在する時に前記FSA方式キャッシュと前記Nウェイセットアソシアティブキャッシュの両方がミスヒットし

た前記場合には、前記NウェイセットアソシアティブキャッシュのN本のミスヒットしたキャッシュラインの中から前記LRU値により指定されたキャッシュラインが選択され、その選択されたキャッシュラインに格納されていたデータが前記FSA方式キャッシュの前記非活性なキャッシュラインのうちの一つに転送され、前記両方のキャッシュのミスヒットにより前記外部メモリからフェッチされたデータは前記FSA方式キャッシュへのデータ転送元となった前記Nウェイセットアソシアティブキャッシュの前記選択されたキャッシュラインに書き込まれるようにし、前記Nウェイセットアソシアティブキャッシュに前記インデックスに対応する非活性なキャッシュラインが存在せず前記FSA方式キャッシュが活性なキャッシュラインで一杯な時に前記FSA方式キャッシュと前記Nウェイセットアソシアティブキャッシュの両方がミスヒットした前記場合には、前記NウェイセットアソシアティブキャッシュのN本のミスヒットしたキャッシュラインの中から前記LRU値により指定されたキャッシュラインが選択され、その選択されたキャッシュラインに格納されていたデータが前記FSA方式キャッシュの前記最小のキャッシュヒットカウント値を持つ前記キャッシュラインのうちの一つに転送され、前記両方のキャッシュのミスヒットにより前記外部メモリからフェッチされたデータは前記FSA方式キャッシュへのデータ転送元転送元となった前記Nウェイセットアソシアティブキャッシュの前記選択されたキャッシュラインに書き込まれるようにし、前記Nウェイセットアソシアティブキャッシュに前記インデックスに対応する非活性なキャッシュラインが1以上存在する時に前記FSA方式キャッシュと前記Nウェイセットアソシアティブキャッシュの両方がミスヒットした前記場合には、前記両方のキャッシュのミスヒットにより前記外部メモリからフェッチされたデータは前記Nウェイセットアソシアティブキャッシュの前記インデックスに対応する前記非活性なキャッシュラインのうちの一つのキャッシュラインに書き込まれるようにしたものである。

【0027】請求項9記載のキャッシュシステムは、請求項1から8のうちのいずれか1項に記載のキャッシュシステムにおいて、前記FSA方式キャッシュと前記FSA方式以外のキャッシュからなる一次キャッシュに付加して設けられた二次キャッシュをさらに備えるようにしたものである。

【0028】請求項10記載のキャッシュ処理方法は、キャッシュヒット率の高いキャッシュラインの保持が可能なFSA (フルセットアソシアティブ) 方式キャッシュと前記FSA方式キャッシュと同時にキャッシュヒット判定のためのタグ比較を行うFSA方式以外のキャッシュとの組み合わせを用い、CPU (Central Processing Unit)、MPU (Micro Processor Unit) 等のデータ処理ユニット

の、メインメモリ等の外部メモリへのアクセス回数を削減するためのキャッシュ処理方法において、入力アドレスデータから抽出された第1のタグが前記FSA方式キャッシュの活性なキャッシュラインに格納されていた複数の第1のタグと比較され、前記第1のタグの一致があった場合に前記FSA方式キャッシュがキャッシュヒットしたと判定される第1のタグ比較工程と、前記第1のタグ比較工程と同時に行われ、前記入力アドレスデータから抽出された第2のタグが前記FSA方式以外のキャッシュの前記入力アドレスデータから抽出されたインデックスに対応する1以上の活性なキャッシュラインに格納されていた第2のタグと比較され、前記第2のタグの一致があった場合に前記FSA方式以外のキャッシュがキャッシュヒットしたと判定される第2のタグ比較工程と、前記FSA方式キャッシュの各キャッシュラインに起きたキャッシュヒットがカウントされ、前記FSA方式キャッシュの各キャッシュラインに対応して設けられたキャッシュヒットカウント格納手段に格納された各キャッシュライン毎のキャッシュヒットカウント値が管理し更新されるキャッシュヒットカウント管理工程と、前記第1のタグ比較工程において前記FSA方式キャッシュがキャッシュヒットした場合に、前記FSA方式キャッシュの前記一致した第1のタグに対応するキャッシュラインに格納されていたデータが、前記データ処理ユニットによりキャッシュデータとして読み込まれる第1のキャッシュデータ読み込み工程と、前記第2のタグ比較工程において前記FSA方式以外のキャッシュがキャッシュヒットした場合に、前記FSA方式以外のキャッシュの前記インデックスと前記一致した第2のタグに対応するキャッシュラインに格納されていたデータが、前記データ処理ユニットによりキャッシュデータとして読み込まれる第2のキャッシュデータ読み込み工程と、前記FSA方式以外のキャッシュに前記インデックスに対応する非活性なキャッシュラインが存在せず前記FSA方式キャッシュに1以上の非活性なキャッシュラインが存在する時に前記FSA方式キャッシュと前記FSA方式以外のキャッシュの両方がミスヒットした場合に、前記FSA方式以外のキャッシュのミスヒットした前記インデックスに対応するあるキャッシュラインに格納されていたデータが前記FSA方式キャッシュの前記非活性なキャッシュラインのうちの一つに転送され、前記両方のキャッシュのミスヒットにより前記外部メモリからフェッチされたデータが前記FSA方式キャッシュへのデータ転送元となった前記FSA方式以外のキャッシュの前記キャッシュラインに書き込まれる第1のデータ転送工程と、前記FSA方式以外のキャッシュに前記インデックスに対応する非活性なキャッシュラインが存在せず前記FSA方式キャッシュが活性なキャッシュラインで一杯な時に前記FSA方式キャッシュと前記FSA方式以外のキャッシュの両方がミスヒットした場合に、前記F

SA方式以外のキャッシュのミスヒットした前記インデックスに対応するあるキャッシュラインに格納されていたデータが前記FSA方式キャッシュの最小のキャッシュヒットカウント値を持つキャッシュラインのうちの一つに転送され、前記両方のキャッシュのミスヒットにより前記外部メモリからフェッチされたデータが前記FSA方式キャッシュへのデータ転送元となった前記FSA方式以外のキャッシュの前記キャッシュラインに書き込まれる第2のデータ転送工程と、前記FSA方式以外のキャッシュに前記インデックスに対応する非活性なキャッシュラインが存在する時に前記FSA方式キャッシュと前記FSA方式以外のキャッシュの両方がミスヒットした場合に、前記両方のキャッシュのミスヒットにより前記外部メモリからフェッチされたデータが前記FSA方式以外のキャッシュの前記インデックスに対応する非活性なキャッシュラインに書き込まれる第3のデータ転送工程とを備えるようにしたものである。

【0029】請求項11記載のキャッシュ処理方法は、請求項10記載のキャッシュ処理方法の前記キャッシュヒットカウント管理工程において、前記FSA方式キャッシュのあるキャッシュラインがキャッシュヒットした場合には、そのキャッシュラインの前記キャッシュヒットカウント値が1つインクリメントされ、前記FSA方式キャッシュと前記FSA方式以外のキャッシュの両方がミスヒットした場合には、前記FSA方式キャッシュの全キャッシュラインの前記キャッシュヒットカウント値が一斉に1つデクリメントされるようにしたものである。

【0030】請求項12記載のキャッシュ処理方法は、請求項10記載のキャッシュ処理方法の前記キャッシュヒットカウント管理工程において、前記FSA方式キャッシュのあるキャッシュラインがキャッシュヒットした場合には、そのキャッシュラインの前記キャッシュヒットカウント値が1つインクリメントされ、前記FSA方式キャッシュがミスヒットした場合には、前記FSA方式キャッシュの全キャッシュラインの前記キャッシュヒットカウント値が一斉に1つデクリメントされるようにしたものである。

【0031】請求項13記載のキャッシュ処理方法は、請求項10記載のキャッシュ処理方法の前記キャッシュヒットカウント管理工程において、前記FSA方式キャッシュのあるキャッシュラインがキャッシュヒットした場合には、そのキャッシュラインの前記キャッシュヒットカウント値が1つインクリメントされるようにしたものである。

【0032】請求項14記載のキャッシュ処理方法は、請求項10から13のうちのいずれか1項に記載のキャッシュ処理方法において、前記FSA方式以外のキャッシュとして、ダイレクトマッピング方式キャッシュを用いるようにしたものである。

10

20

30

40

50

【0033】請求項15記載のキャッシュ処理方法は、請求項10から13のうちのいずれか1項に記載のキャッシュ処理方法において、前記FSA方式以外のキャッシュとして、Nウェイセットアソシアティブキャッシュ(N=2, 4, 8, ...)を用いるようにしたものである。

【0034】請求項16記載のキャッシュ処理方法は、請求項10から15のうちのいずれか1項に記載のキャッシュ処理方法において、前記FSA方式キャッシュの各キャッシュラインに生じた最新のキャッシュヒットの日時に関するデータを前記各キャッシュラインに対応して設けられたキャッシュヒット日時格納手段に格納するキャッシュヒット日時格納工程をさらに備えるようにし、前記第2のデータ転送工程においては、前記キャッシュヒット日時格納手段に格納された前記データに基づいて前記FSA方式キャッシュの前記最小のキャッシュヒットカウント値を持つ前記キャッシュラインの中から最近のキャッシュヒットが最も古いキャッシュラインが選択され、その選択されたキャッシュラインが前記FSA方式以外のキャッシュから前記FSA方式キャッシュへの前記データ転送の転送先として指定されるようにしたものである。

【0035】請求項17記載のキャッシュ処理方法は、請求項15記載のキャッシュ処理方法において、各インデックスに対応するN本のキャッシュラインの中で最近のキャッシュヒットが最も古いキャッシュラインを示すためのLRU(Least Recently Used)値を各インデックスに対応して設けられたLRU格納手段に格納するLRU格納工程をさらに備えるようにし、前記第1のデータ転送工程においては、前記NウェイセットアソシアティブキャッシュのN本のミスヒットしたキャッシュラインの中から前記LRU値により指定されたキャッシュラインが選択され、その選択されたキャッシュラインに格納されていたデータが前記FSA方式キャッシュの前記非活性なキャッシュラインのうちの一つに転送され、前記両方のキャッシュのミスヒットにより前記外部メモリからフェッチされたデータは前記FSA方式キャッシュへのデータ転送元となった前記Nウェイセットアソシアティブキャッシュの前記選択されたキャッシュラインに書き込まれるようにし、前記第2のデータ転送工程においては、前記NウェイセットアソシアティブキャッシュのN本のミスヒットしたキャッシュラインの中から前記LRU値により指定されたキャッシュラインが選択され、その選択されたキャッシュラインに格納されていたデータが前記FSA方式キャッシュの前記最小のキャッシュヒットカウント値を持つ前記キャッシュラインのうちの一つに転送され、前記両方のキャッシュのミスヒットにより前記外部メモリからフェッチされたデータは前記FSA方式キャッシュへのデータ転送元となった前記Nウェイセットアソシアティブキャッシュ

のキャッシュの前記選択されたキャッシュラインに書き込まれるようにし、前記第3のデータ転送工程においては、前記両方のキャッシュのミスヒットにより前記外部メモリからフェッチされたデータは前記Nウェイセットアソシアティブキャッシュの前記インデックスに対応する前記非活性なキャッシュラインのうちの一つのキャッシュラインに書き込まれるようにしたものである。

【0036】請求項18記載のキャッシュ処理方法は、請求項10から17のうちのいずれか1項に記載のキャッシュ処理方法において、前記FSA方式キャッシュと前記FSA方式以外のキャッシュの両方がミスヒットした場合に前記FSA方式キャッシュと前記FSA方式以外のキャッシュからなる一次キャッシュに付加する形で設けられた二次キャッシュへのアクセスが行われる二次キャッシュアクセス工程をさらに備えるようにしたものである。

【0037】請求項19から27に記載の機械読み取り可能な記録媒体は、請求項10から18に記載のキャッシュ処理方法に従うキャッシュ処理をコンピュータに実行させるためのプログラムを格納した機械読み取り可能な記録媒体である。

#### 【0038】

【発明の実施の形態】次に、本発明の実施の形態について図面を参照して詳細に説明する。

【0039】実施の形態1. 図1は本発明の実施の形態1によるキャッシュシステムのキャッシュ部15の構成を示すブロック図であり、図2は本発明の実施の形態1によるキャッシュシステムが適用されたコンピュータシステムの例を示すブロック図である。

【0040】図2のコンピュータシステムは、CPU(Central Processing Unit)50、二次キャッシュ20、およびメインメモリ21を備えており、二次キャッシュ20はCPU50の外部に設けられている。二次キャッシュ20は、例えば、タグRAMとデータRAMとを備えたシンクロナスSRAMや、ページROMを備えたメモリによって実現されている。二次キャッシュ20の種類(フルセットアソシアティブ方式であるか否か等)は特に限定されない。図2のコンピュータシステムには、CPU50、二次キャッシュ20、メインメモリ21等を接続するアドレスバス1とデータバス2が設けられており、メインメモリ21は、CPU50の外部のアドレスバス1とデータバス2とに、二次キャッシュ20と共に接続されている。

【0041】CPU50は、バスインタフェース(I/F)部3、書込バッファ4、コントロールレジスタ5、MMU(Memory Management Unit)レジスタ6、ALU(arithmetic logic operation unit)7、GPR(General Purpose Register)レジスタ8、マルチプライヤ9、バレルシフタ10、プロ



グラムカウンタ 11、クロックジェネレータ 12、キャッシュ部 15等を有している。このキャッシュ部 15は、本実施の形態のキャッシュシステムの一次キャッシュとして CPU 50 に設けられているもので、このキャッシュ部 15 もアドレスバス 1 とデータバス 2 とに接続されている。

【0042】クロックジェネレータ 12 は、与えられたマスタクロック信号の周波数を逡倍して本コンピュータシステムのクロック信号を生成し、そのクロック信号をコンピュータシステムの各部品に供給する。書込バッファ 4 は、メインメモリ 21 や二次キャッシュ 20 等に書き込まれるシステムデータを一時的に格納するバッファであり、バス 1 / F 部 3 を介してアドレスバス 1 とデータバス 2 とに接続されている。システムデータの書込は、バス 1 / F 部 3 に供給される制御データに従って行われる。コントロールレジスタ 5 と MMU レジスタ 6 も同様にアドレスバス 1 とデータバス 2 に接続されている。コントロールレジスタ 5 は、クロックジェネレータ 12 の周波数逡倍数などの、CPU 50 の基本設定に関するデータを格納し、MMU レジスタ 6 は仮想アドレス-物理アドレス間のアドレス変換用のデータを格納する。ALU 7、GPR レジスタ 8、マルチプライヤ 9、バレルシフタ 10 およびプログラムカウンタ 11 を含む各命令の実行のためのユニットも、アドレスバス 1 とデータバス 2 に接続されている。なお、この部品 7 ~ 11 を含むユニットを以降、命令実行ユニットと呼ぶ。

【0043】図 1 を参照すると、本実施の形態のキャッシュシステムの一次キャッシュとしてのキャッシュ部 15 は、フルセットアソシアティブ (FSA) 方式キャッシュテーブル TA、ダイレクトマッピング (DM) 方式

キャッシュテーブル TB、コンパレータ 30A および 30B、AND ゲート 31A および 31B、OR ゲート 32、およびデータセクタ 33 を備えている。

【0044】図 1 に示すように、本実施の形態の一次キャッシュすなわちキャッシュ部 15 は、フルセットアソシアティブ (FSA) 方式キャッシュメモリとダイレクトマッピング (DM) 方式キャッシュメモリの組み合わせによって構成される。なお、図 1 においては 1 個のコンパレータ 30A と 1 個の AND ゲート 31A のみが示されているが、具体的には、FSA 方式キャッシュテーブル TA の各キャッシュライン毎にコンパレータ 30A と AND ゲート 31A が設けられ、全キャッシュラインの AND ゲート 31A の出力の論理和 (OR) が図示しない 1 以上の OR ゲートを用いて取られる。

【0045】図 1 の上部には、入力アドレスデータの例として「00000040」(16 進) が示されている。この入力アドレスデータは、タグ # 2、インデックス、およびオフセットを含んでいる。このタグ # 2 は、DM 方式キャッシュテーブル TB に関して使用されるタグであり、FSA 方式キャッシュテーブル TA に関して

使用されるもう一つのタグであるタグ # 1 が、タグ # 2 とインデックスとの和として定義される (タグ # 1 = タグ # 2 + インデックス)。例えば入力アドレスデータ「00000040」の場合には、タグ # 2 は「000000」(例えば 16 進数の入力アドレスデータの最初の 5 桁)、インデックスは「04」(例えば 16 進数の入力アドレスデータの次の 2 桁)、タグ # 1 は「0000004」(例えば 16 進数の入力アドレスデータの最初の 7 桁)、オフセットは「0」(例えば 16 進数の入力アドレスデータの最後の 1 桁) となる。

【0046】FSA 方式キャッシュテーブル TA の各キャッシュラインには、そのキャッシュラインに対して発生したアクセス (キャッシュヒット) の数を示すためのアクセスカウンタ領域 AC が設けられている。後に説明するような方法によりアクセスカウンタ領域 AC を用いて各キャッシュラインのアクセス頻度を管理することにより、FSA 方式キャッシュテーブル TA における高アクセス頻度 (高キャッシュヒット率) のキャッシュラインの保持が実現される。なお、FSA 方式キャッシュテーブル TA の全キャッシュラインのアクセスカウンタ領域 AC を管理するためのアクセスカウンタが、例えば、前述の部品 7 ~ 11 を有した命令実行ユニットにより実現される。また、この実施の形態のキャッシュシステムの動作を制御するプログラムが、図示しない ROM 等の記録媒体に格納されており、このプログラムに従って後に詳細に述べるキャッシュシステムの動作が行われる。

【0047】コンパレータ 30A は、入力アドレスデータから抽出されたタグ # 1 と、FSA 方式キャッシュテーブル TA の各キャッシュラインに格納されていたタグ # 1 とを比較する。タグ # 1 の一致が有った場合には、コンパレータ 30A はハイレベル (1) の比較信号を AND ゲート 31A に出力する。AND ゲート 31A には、その一致したタグ # 1 に対応するキャッシュラインが活性か非活性かを示す活性化ビット V1 (1/0) が FSA 方式キャッシュテーブル TA から入力されており、AND ゲート 31A はコンパレータ 30A からの前記比較信号と FSA 方式キャッシュテーブル TA からの前記活性化ビット V1 が共に「1」であればキャッシュヒット信号 (ハイレベル (1)) を出力する。なお、前述のように、図 1 には 1 個のコンパレータ 30A と 1 個の AND ゲート 31A として示しているが具体的には FSA 方式キャッシュテーブル TA の各キャッシュライン毎にコンパレータ 30A と AND ゲート 31A が設けられ、全キャッシュラインの AND ゲート 31A の出力の論理和 (OR) が取られる。

【0048】このような FSA 方式キャッシュにおけるタグ比較と並行して、DM 方式キャッシュにおけるコンパレータ 30B は、入力アドレスデータから抽出されたタグ # 2 と、入力アドレスデータから抽出されたインデックスに対応する DM 方式キャッシュテーブル TB のキ

10

20

30

40

50

キャッシュラインに格納されていたタグ#2とを比較する。タグ#2が一致した場合には、コンパレータ30Bはハイレベル(1)の比較信号をANDゲート31Bに出力する。ANDゲート31Bには、その一致したタグ#2に対応するキャッシュラインが活性か非活性かを示す活性化ビットV2(1/0)がDM方式キャッシュテーブルTBから入力されており、ANDゲート31Bはコンパレータ30Bからの前記比較信号とDM方式キャッシュテーブルTBからの前記活性化ビットV2が共に「1」であればキャッシュヒット信号(ハイレベル(1))を出力する。

【0049】ORゲート32は、ANDゲート31AまたはANDゲート31Bからハイレベル(1)のキャッシュヒット信号が入力されている場合に、活性化信号(ハイレベル(1))を出力する。ORゲート32からハイレベル(1)の活性化信号が入力された場合、データセクタ33は、FSA方式キャッシュテーブルTAまたはDM方式キャッシュテーブルTBの前記ヒットしたキャッシュラインに格納されていたデータの一部を入力アドレスデータのオフセットに基づいて選択し、その選択したデータをCPU50の図示しない命令デコーダまたはGPRレジスタ8に送る。

【0050】なお、この実施の形態においては、キャッシュヒットが起こる際にはFSA方式キャッシュテーブルTAまたはDM方式キャッシュテーブルTBの一方で起きる。すなわち、FSA方式キャッシュテーブルTAとDM方式キャッシュテーブルTBの両方でキャッシュヒットが起きる場合はない。

【0051】図1に示すFSA方式キャッシュテーブルTA(FSA方式キャッシュメモリ)は、DM方式キャッシュテーブルTB(DM方式キャッシュメモリ)と比較して消費電力が大きく、かつ、回路構成が複雑であるものの、ヒット率の高いキャッシュラインを保持させることが可能である。一方、DM方式キャッシュテーブルTB(DM方式キャッシュメモリ)は、小さな回路規模の割に高速アクセスが可能であるが、ヒット率が容易に悪化してしまう欠点がある。この実施の形態のキャッシュシステムの一次キャッシュ(キャッシュ部15)は、このような特徴を有するFSA方式キャッシュテーブルTAとDM方式キャッシュテーブルTBとを組み合わせることにより構成されている。

【0052】以下、この発明の実施の形態1によるキャッシュシステムの動作について詳細に説明する。図3は、本発明の実施の形態1によるキャッシュシステムの動作を示すフローチャートである。なお、この実施の形態においては、一次キャッシュとしてのキャッシュ部15がミスヒットした場合にメインメモリ21または二次キャッシュ20からフェッチされたデータは、DM方式キャッシュテーブルTBに格納、登録され、FSA方式キャッシュテーブルTAには登録されない。キャッシュ

部15がミスヒットした場合(すなわち、FSA方式キャッシュテーブルTAと、入力アドレスデータから抽出されたインデックスに対応するDM方式キャッシュテーブルTBのキャッシュラインとの両方がミスヒットした場合)、DM方式キャッシュテーブルTBのそのインデックスに対応するキャッシュラインに格納されていたデータは、DM方式キャッシュテーブルTBにおいて削除(または非活性化)され、FSA方式キャッシュテーブルTAに転送される。

10 【0053】CPU50はプログラムカウンタ11から供給される入力アドレスデータに従って命令フェッチ処理を開始する(ステップS10)。FSA方式キャッシュメモリにおいて、図1に示すコンパレータ30A(具体的にはFSA方式キャッシュテーブルTAの各キャッシュラインに対応して設けられている複数のコンパレータ30A)が入力アドレスデータから抽出されたタグ#1(入力アドレスデータのオフセット以外の部分)と、FSA方式キャッシュテーブルTAの各キャッシュラインに格納されていたタグ#1とを比較する。タグ#1の一致があった場合は(ステップS11:Yes)、コンパレータ30Aはハイレベル(1)の前記比較信号をANDゲート31Aに出力する。コンパレータ30Aの出力と、FSA方式キャッシュテーブルTAのヒットしたタグ#1に対応するキャッシュラインの活性化ビットV1が共に「1」である場合には(ステップS12:Yes)、図1に示すANDゲート31A(具体的にはFSA方式キャッシュテーブルTAの各キャッシュラインに対応して設けられている複数のANDゲート31Aとその複数のANDゲート31Aの出力の論理和(OR)を取るための1以上のORゲート)がハイレベル(1)の前記キャッシュヒット信号をORゲート32に出力する。このキャッシュヒット信号は、FSA方式キャッシュテーブルTAで有効なキャッシュヒットが生じたことを示す。抽出されたタグ#1がFSA方式キャッシュテーブルTAのどのタグ#1とも一致しなかった場合(ステップS11:No)、またはヒットしたキャッシュラインの活性化ビットV1が「0」である場合(ステップS12:No)には、ANDゲート31Aの出力は「0」となる(ステップS13)。

40 【0054】同時に、キャッシュヒットの有無の判定のためのタグ比較がDM方式キャッシュメモリにおいても行われる。DM方式キャッシュメモリにおいては、インデックス(16進数の入力アドレスデータの最初の5桁であるタグ#2に続く2桁)が入力アドレスデータより抽出され、DM方式キャッシュテーブルTBのそのインデックスに対応するキャッシュラインが選択、指定される(ステップS14)。図1に示すコンパレータ30Bは、入力アドレスデータから抽出されたタグ#2(入力アドレスデータのインデックスとオフセット以外の部分)と、DM方式キャッシュテーブルTBの前記インデ

ックスに対応する選択されたキャッシュラインに格納されていたタグ#2とを比較する。タグ#2が一致した場合（ステップS15:Yes）、コンパレータ30Bはハイレベル（1）の比較信号をANDゲート31Bに出力する。コンパレータ30Bの出力とヒットしたキャッシュラインの活性化ビットV2が共に「1」である場合には（ステップS16:Yes）、ANDゲート31Bはハイレベル（1）のキャッシュヒット信号をORゲート32に出力する。このキャッシュヒット信号は、DM方式キャッシュテーブルTBで有効なキャッシュヒットが生じたことを示す。タグ#2が一致しなかった場合（ステップS15:No）、またはヒットしたキャッシュラインの活性化ビットV2が「0」である場合（ステップS16:No）には、ANDゲート31Bの出力は「0」となる（ステップS17）。

【0055】図1に示すORゲート32は、ANDゲート31Aからのキャッシュヒット信号またはANDゲート31Bからのキャッシュヒット信号が「1」であれば、ハイレベル（1）の前記活性化信号をデータセクタ33に出力する。それ以外の場合（両方からのキャッシュヒット信号が共に「0」である場合）は、ORゲート32はローレベル（0）の活性化信号をデータセクタ33に出力する。なお、前述のように、この実施の形態においては、FSA方式キャッシュテーブルTAとDM方式キャッシュテーブルTBの両方でキャッシュヒットが起きる場合はない。すなわち、ANDゲート31AとANDゲート31Bの両方からのキャッシュヒット信号が「1」である場合はない。

【0056】ORゲート32から出力される活性化信号が「0」である場合（キャッシュ部15がミスヒットしたことを示す）（ステップS18:No）、CPU50は二次キャッシュ20を介してメインメモリ21にアクセスし、メインメモリ21から（二次キャッシュ20がミスヒットした場合）または二次キャッシュ20から（二次キャッシュ20がキャッシュヒットした場合）必要なデータをフェッチする。この場合、DM方式キャッシュテーブルTBのミスヒットした前記インデックスに対応するキャッシュラインに格納されていたデータ（タグ#2およびデータ）はFSA方式キャッシュテーブルTAに転送され（DM方式キャッシュテーブルTBにおいては削除または非活性化される）、メインメモリ21または二次キャッシュ20からフェッチされたデータがDM方式キャッシュテーブルTBのそのキャッシュラインに格納される（ステップS19）。なお、この際にDM方式キャッシュテーブルTBにインデックスに対応する非活性な（活性化ビットV2が「0」の）キャッシュラインが存在した場合、前記のデータのFSA方式キャッシュテーブルTAへの転送は行われず、メインメモリ21または二次キャッシュ20からフェッチされたデータはDM方式キャッシュテーブルTBのインデックスに

対応する非活性なキャッシュラインに格納される（ステップS19）。その後、CPU50はDM方式キャッシュテーブルTBの前記キャッシュラインから前記データを読み出し、そのデータ（データまたは命令）に関する処理を行う（ステップS21）。

【0057】ORゲート32から出力される活性化信号が「1」である場合（キャッシュ部15がキャッシュヒットしたことを示す）（ステップS18:Yes）、データセクタ33が、FSA方式キャッシュテーブルTAまたはDM方式キャッシュテーブルTBのヒットしたキャッシュラインに格納されていたデータの一部を入力アドレスデータのオフセットに基づいて選択、抽出し（ステップS20）、CPU50がその選択されたデータ（データまたは命令）に関する処理を行う（ステップS21）。

【0058】以下において、FSA方式キャッシュテーブルTAおよびDM方式キャッシュテーブルTBの動作について詳細に説明する。

【0059】初期化（図2のコンピュータシステムがリセットまたは電源ONされた際）において、FSA方式キャッシュテーブルTAとDM方式キャッシュテーブルTBのすべてのキャッシュラインが非活性化される。すなわち、全キャッシュラインの活性化ビットV1、V2が「0」にリセットされる。その後、FSA方式キャッシュテーブルTAまたはDM方式キャッシュテーブルTBのあるキャッシュラインにデータが書き込みまたは転送される毎に、そのキャッシュラインが活性化される、すなわちそのキャッシュラインの活性化ビットV1またはV2が「0」から「1」に書き替えられる。

【0060】キャッシュ部15がミスヒットした場合（すなわちFSA方式キャッシュテーブルTAと、DM方式キャッシュテーブルTBのインデックスに対応するキャッシュラインとが共にミスヒットした場合）、DM方式キャッシュテーブルTBのそのキャッシュラインに格納されていたデータ（データとタグ#2）が、FSA方式キャッシュテーブルTAの非活性なキャッシュラインに転送される。この転送により、DM方式キャッシュテーブルTBのそのキャッシュラインは非活性化され（すなわち、そのキャッシュラインに格納されていたデータが非活性化または実質的に削除され）、FSA方式キャッシュテーブルTAの前記キャッシュラインは活性化される。DM方式キャッシュテーブルTBの非活性化されたキャッシュラインには、メインメモリ21または二次キャッシュ20からフェッチされたデータが書き込まれ、これによりそのキャッシュラインは再度活性化される。なお前述のように、DM方式キャッシュテーブルTBにインデックスに対応する非活性なキャッシュラインが存在した場合、前記のデータのFSA方式キャッシュテーブルTAへの転送は行われず、メインメモリ21または二次キャッシュ20からフェッチされたデータは

DM方式キャッシュテーブルTBのインデックスに対応する非活性なキャッシュラインに書き込まれる。

【0061】FSA方式キャッシュテーブルTAの活性状態のキャッシュラインは、各キャッシュラインのアクセスカウント領域ACに格納されるアクセスカウント値を用いて管理、制御される。なおこの実施の形態におけるアクセスカウント領域ACのビット数は、例えば2ビットに設定される。FSA方式キャッシュテーブルTAのあるキャッシュラインにデータが書き込まれた際には、そのキャッシュラインのアクセスカウント値が

「0」に、前述の図示しないアクセスカウンタによりリセットされる。この図示しないアクセスカウンタは、キャッシュヒットがFSA方式キャッシュテーブルTAのあるキャッシュラインに生じる毎にそのキャッシュラインのアクセスカウント値を1ずつインクリメントする。キャッシュ部15がミスヒットした場合（すなわちFSA方式キャッシュテーブルTAと、DM方式キャッシュテーブルTBのインデックスに対応するキャッシュラインとが共にミスヒットした場合）、前記アクセスカウンタは、FSA方式キャッシュテーブルTAの全キャッシュ\*20

AC      キャッシュラインの実質的アクセス数（キャッシュヒット数）

0	0回以下
1	1回
2	2回
3	3回以上

【0064】FSA方式キャッシュテーブルTAに1以上の非活性なキャッシュラインが存在する場合、そのキャッシュラインは前述のように、DM方式キャッシュテーブルTBのミスヒットしたキャッシュラインからFSA方式キャッシュテーブルTAへのデータ転送の際の転送先とみなされる。従って、（DM方式キャッシュテーブルTBにインデックスに対応する非活性なキャッシュラインが存在せず）FSA方式キャッシュテーブルTAに1以上の非活性なキャッシュラインが存在する時にキャッシュ部15がミスヒットした場合、DM方式キャッシュテーブルTBのミスヒットしたキャッシュラインに格納されていたデータ（データとタグ#2）はFSA方式キャッシュテーブルTAの非活性なキャッシュラインの内の一つに転送される。

【0065】一方、FSA方式キャッシュテーブルTAが活性状態のキャッシュラインで一杯の場合には、FSA方式キャッシュテーブルTAの最小のアクセスカウント値を持つキャッシュライン（1本またはそれ以上）が、DM方式キャッシュテーブルTBのミスヒットしたキャッシュラインからFSA方式キャッシュテーブルTAへのデータの転送の際の転送先とみなされる。従って、（DM方式キャッシュテーブルTBにインデックスに対応する非活性なキャッシュラインが存在せず）FSA方式キャッシュテーブルTAが活性状態のキャッシュラインで一杯の時にキャッシュ部15がミスヒットした

\* ユーラインのアクセスカウント値を一斉に1ずつデクリメントする。この実施の形態におけるこのアクセスカウント値の最小値と最大値は、十進数で0と3となる。すなわち、アクセスカウント値の0未満へのデクリメントおよび4以上へのインクリメントは行われない。

【0062】なお、上記においてはキャッシュ部15がミスヒットした際に前記のアクセスカウント値の一斉のデクリメントが行われるものとしたが、FSA方式キャッシュテーブルTAがミスヒットした際（すなわちFSA方式キャッシュテーブルTAのすべてのキャッシュラインがミスヒットした際）にアクセスカウント値の一斉のデクリメントが行われるものとすることも可能である。この場合、DM方式キャッシュテーブルTBがキャッシュヒットしてFSA方式キャッシュテーブルTAがミスヒットした場合も前記アクセスカウント値の一斉のデクリメントが行われる点が上記とは異なり、このため、アクセスカウント値のデクリメントがより高い確率で行われることとなる。

【0063】このようなアクセスカウント値をまとめると、以下ようになる。

場合、DM方式キャッシュテーブルTBのミスヒットしたキャッシュラインに格納されていたデータ（データとタグ#2）はFSA方式キャッシュテーブルTAの最小のアクセスカウント値を持つキャッシュラインの一つに転送される。この場合の転送先は、最小カウント値を持つ1本またはそれ以上のキャッシュラインからランダムに選択してもよいが、最近のキャッシュヒットが最も古いキャッシュラインをその中から選択してそのキャッシュラインを転送先とみなすことも可能である。そのような最近のキャッシュヒットが最も古いキャッシュラインの選択は、例えば、アクセス日時を格納するための領域をFSA方式キャッシュテーブルTAの各キャッシュラインに設けることによって実現できる。

【0066】また、DM方式キャッシュテーブルTBにインデックスに対応する非活性なキャッシュラインが存在する状態でキャッシュ部15がミスヒットした場合には、メインメモリ21または二次キャッシュ20からフェッチされたデータは、DM方式キャッシュテーブルTBのそのインデックスに対応する非活性なキャッシュラインに書き込まれる。この場合、前述のDM方式キャッシュテーブルTBからFSA方式キャッシュテーブルTAへのデータ転送は行われない。

【0067】以下において、キャッシュ処理の実行時間に関して説明する。

【0068】図4は、CPU50によって実行されるプ

プログラムの一例を示す模式図である。なお、図4は従来の技術の説明において参照した図11と同一の図である。図4を参照すると、このプログラムは、実行されるべき命令(1), (2), ...を有している。CPU50はまず、プログラムカウンタ11から供給される、命令(1)に関する入力アドレスデータを参照する。最初の命令(1)に対応する入力アドレスデータ「00000040」は、この命令(1)がメインメモリのアドレス「00000040」に予め格納されていることを示す。前述のようにこの入力アドレスデータ「00000040」は、タグ#2(00000)、インデックス(04)およびオフセット(0)(およびタグ#1(0000004))を含んでいる。このタグ#1、タグ#2、インデックスおよびオフセットを含んだ入力アドレスデータがキャッシュ部15に供給され、キャッシュヒット判定のための処理が前述のようにして行われる。CPU50はこのキャッシュヒット判定処理の結果に従い、必要なデータ(命令(1))をFSA方式キャッシュテーブルTA、DM方式キャッシュテーブルTBまたはメインメモリ21(または二次キャッシュ20)から入手し、命令(1)を実行する。その後は、同様な処理が続く命令(2), (3), ...に対して行われる。

【0069】図5(a)および(b)は、この実施の形態のキャッシュシステムを用いるCPU50が図4のプログラムを2回行う際のアクセス時間を示す模式図であり、図5(a)は1回目のプログラム実行の場合を、図5(b)は2回目のプログラム実行の場合を示している。図6は、図4のプログラムの命令(5)の実行直後のFSA方式キャッシュテーブルTAとDM方式キャッシュテーブルTBの状態を示す模式図である。

【0070】なお、以下のプログラム実行時間に関する説明は、従来技術における説明と同様の仮定の下に行う。すなわち、FSA方式キャッシュテーブルTAとDM方式キャッシュテーブルTBの各キャッシュラインのデータ格納領域の長さは4ワード(16バイト)とし、図4のプログラムの各命令の長さは1ワード(4バイト)とする(すなわち、FSA方式キャッシュテーブルTAとDM方式キャッシュテーブルTBの各キャッシュラインのデータ格納領域に4個の命令が格納されるものとする)。データ(命令)をメインメモリからフェッチするのに必要なアクセス時間は、最初の1ワードに100ns、続く3ワードの各々にそれぞれ30nsとする。従って、4ワード分のデータ(4個の命令)をメインメモリ21からフェッチしてこれらのデータをDM方式キャッシュテーブルTBの1つのキャッシュライン格納するために必要なアクセス時間は、100+30+30+30=190nsとなる。また、CPU50がDM方式キャッシュテーブルTBのキャッシュラインに格納された直後の命令を読み出して実行するのに10nsかかるものとする。また、キャッシュ部15がミスヒッ

トした際にはデータはメインメモリ21からフェッチされる(二次キャッシュ20はキャッシュヒットしない)ものとし、二次キャッシュ20のデータ処理時間は0とする。DM方式キャッシュテーブルTBのキャッシュサイズは例えば4kバイトとする。

【0071】初期化において、FSA方式キャッシュテーブルTAとDM方式キャッシュテーブルTBのすべてのキャッシュラインが非活性化される。すなわち、全キャッシュラインの活性化ビットV1, V2が「0」にリセットされる。この初期化の直後には活性化ビットV1, V2がすべて「0」であるため、FSA方式キャッシュテーブルTAとDM方式キャッシュテーブルTBの中には実質的にはデータが存在しないことになる。このためCPU50は、必要なデータ(最初のインデックスに対応する命令(1)~(4))をメインメモリ21からフェッチし、フェッチした命令(1)~(4)をDM方式キャッシュテーブルTBのインデックス「04」に対応するキャッシュラインに書き込む。このデータ書き込みにより、DM方式キャッシュテーブルTBのインデックス「04」に対応するキャッシュラインは活性化される。すなわち、このキャッシュラインの活性化ビットV2が「1」にセットされる(図1参照)。その後、CPU50はそのキャッシュラインから最初の命令(1)を読み出して実行する。

【0072】図5(a)に示される最初のプログラム実行においては、コンピュータシステムのスタートアップルーチンの後にCPU50が最初の命令(1)を実行するのに100+30+30+30+10=200nsが必要であり、続く命令(2), (3), (4)の実行には各々10nsが必要である(230ns)。その後CPU50は、アドレス「00001030」から「0000103C」に対応する4個の命令を同様に実行する(200+10+10+10=230ns)。

【0073】前述のインデックス「04」に対応する命令(5)をCPU50が実行しようとする時、DM方式キャッシュテーブルTBのインデックス「04」に対応するキャッシュラインには、既に命令(1)とこの命令(1)に対応するタグ「00000」が格納されているため、このインデックス「04」に対応するキャッシュラインにミスヒット(「00001」≠「00000」)が起こる。1回目のプログラム実行においてはFSA方式キャッシュテーブルTAもミスヒットする。このため、DM方式キャッシュテーブルTBのインデックス「04」に対応するキャッシュラインに格納されていたこの命令(1)とタグ#2(00000)がFSA方式キャッシュテーブルTAの非活性化キャッシュラインに(または最小アクセスカウント値のキャッシュラインに)転送され、メインメモリ21からフェッチされた命令(5)と対応するタグ#2(00001)がDM方式キャッシュテーブルTBのそのインデックス「04」に

対応するキャッシュラインに書き込まれる。このようなミスヒットのため、命令(5)の実行も $100 + 30 + 30 + 30 + 10 = 200 \text{ ns}$ かかる。続く3個の命令の実行にも同様に各々 $10 \text{ ns}$ かかる( $230 \text{ ns}$ )。従って、1回目のプログラム実行における総プログラム実行時間は $230 \times 3 = 690 \text{ ns}$ となる。

【0074】図5(a)の1回目のプログラム実行が終了した時点で、図4のプログラムの全ての命令がFSA方式キャッシュテーブルTAまたはDM方式キャッシュテーブルTBに格納された状態となっている。例えば、2回目のプログラム実行において(前述のインデックス「04」に対応する)命令(1)をCPU50が実行しようとする時、この命令(1)は図6に示すように既にFSA方式キャッシュテーブルTAの方に格納されており、命令(1)の2回目の実行においてミスヒットは発生しない。このようにして、図5(b)の2回目のプログラム実行においては、各命令はその実行に $10 \text{ ns}$ ずつしかかからず、2回目のプログラム実行におけるプログラム実行時間は $10 \times 12 = 120 \text{ ns}$ となる。よって、図4(図11)のプログラムを2回実行するために必要な総プログラム実行時間(アクセス時間)は $690 + 120 = 810 \text{ ns}$ となる。

【0075】前述のように、従来のダイレクトマッピング方式キャッシュを用いた場合には2回のプログラム実行に $1190 \text{ ns}$ の総プログラム実行時間が必要であった。従って、本実施の形態のキャッシュシステムの使用により、 $380 \text{ ns}$ のプログラム実行時間が削減されたこととなる。

【0076】上記のように、本発明の実施の形態1によるキャッシュシステムにおいては、FSA方式キャッシュテーブルTAとDM方式キャッシュテーブルTBを組み合わせたキャッシュ部15が、キャッシュシステムの一次キャッシュとして用いられ、キャッシュヒット判定のためのタグ比較がFSA方式キャッシュテーブルTAとDM方式キャッシュテーブルTBの両方で同時に行われる。このため、一次キャッシュとしてのキャッシュ部15では、高速のタグ比較がより高いキャッシュヒット率で行われ、CPU50の(二次キャッシュ20または)メインメモリ21へのアクセスの回数とデータアクセス時間が削減され、これによりCPU50の高速なプログラム実行が実現される。

【0077】さらに、本キャッシュシステムにおいては、以下のようにしてデータの管理/転送が行われる。FSA方式キャッシュテーブルTAの各キャッシュラインへのアクセス(キャッシュヒット)の回数(アクセスカウンタ値)が、各キャッシュラインのアクセスカウンタ領域ACに格納される。FSA方式キャッシュテーブルTAのあるキャッシュラインにキャッシュヒットが生じた場合、そのキャッシュラインのアクセスカウンタ値が1だけインクリメントされ、キャッシュ部15がミス

ヒットした場合(またはFSA方式キャッシュテーブルTAがミスヒットした場合)、FSA方式キャッシュテーブルTAの全キャッシュラインのアクセスカウンタ値が一斉に1だけデクリメントされる。

【0078】DM方式キャッシュテーブルTBにインデックスに対応する非活性なキャッシュラインが存在せず、FSA方式キャッシュテーブルTAに1以上の非活性なキャッシュラインが存在する時に、キャッシュ部15のミスヒットが発生した場合には、DM方式キャッシュテーブルTBのミスヒットしたキャッシュラインに格納されていたデータ(データとタグ#2)はFSA方式キャッシュテーブルTAの非活性なキャッシュラインのうちの一つに転送され、キャッシュ部15のミスヒットによりメインメモリ21(または二次キャッシュ20)からフェッチされたデータはDM方式キャッシュテーブルTBのその(FSA方式キャッシュテーブルTAヘデータ(データとタグ#2)が転送された)キャッシュラインに書き込まれる。

【0079】DM方式キャッシュテーブルTBにインデックスに対応する非活性なキャッシュラインが存在せず、FSA方式キャッシュテーブルTAが活性なキャッシュラインで一杯な時に、キャッシュ部15のミスヒットが発生した場合には、DM方式キャッシュテーブルTBのミスヒットしたキャッシュラインに格納されていたデータ(データとタグ#2)はFSA方式キャッシュテーブルTAの最小のアクセスカウンタ値を持つキャッシュラインの一つに転送され、キャッシュ部15のミスヒットによりメインメモリ21(または二次キャッシュ20)からフェッチされたデータはDM方式キャッシュテーブルTBのその(FSA方式キャッシュテーブルTAヘデータ(データとタグ#2)が転送された)キャッシュラインに書き込まれる。好ましくは、FSA方式キャッシュテーブルTAの最小のアクセスカウンタ値を持つキャッシュラインの中から最新のキャッシュヒットが最も古いキャッシュラインが選択され、その選択されたキャッシュラインがDM方式キャッシュテーブルTBからFSA方式キャッシュテーブルTAへのデータ転送の際の転送先として指定される。

【0080】DM方式キャッシュテーブルTBにインデックスに対応する非活性なキャッシュラインが存在する時にキャッシュ部15のミスヒットが発生した場合には、キャッシュ部15のミスヒットによりメインメモリ21(または二次キャッシュ20)からフェッチされたデータはDM方式キャッシュテーブルTBのそのインデックスに対応する非活性なキャッシュラインに書き込まれる。

【0081】このようなデータ管理/転送により、FSA方式キャッシュテーブルTAが保持するキャッシュラインの更新が各キャッシュラインのアクセスカウンタ値(実質的ヒットカウンタ)を基にして行われる。これに

10

20

30

40

50

より、FSA方式キャッシュテーブルTAにおける高ヒット率のキャッシュラインの保持性能が高められ、キャッシュ部15のキャッシュヒット率が向上する。さらに、データのFSA方式キャッシュテーブルTAとDM方式キャッシュテーブルTBにおける重複が完全に防止される。すなわち、同一のデータがFSA方式キャッシュテーブルTAとDM方式キャッシュテーブルTBの両方に格納される場合がない。従って、FSA方式キャッシュテーブルTAとDM方式キャッシュテーブルTBのメモリ空間を完全に有効に利用することができ、限られたデータ格納容量のキャッシュ部15中に格納できるキャッシュデータの量を最大にすることができる。これにより、キャッシュ部15のキャッシュヒット率をより有効に増加させてCPU50のメインメモリ21（または二次キャッシュ20）へのアクセス回数を最小化することができ、CPU50のプログラム実行時間削減を極めて有効に行うことができる。

【0082】なお、実施の形態1のキャッシュシステムは二次キャッシュ20を備えるものとして説明したが、上記の実施の形態1の各効果は、図7に示す例のようにキャッシュシステムに二次キャッシュ20が設けられない場合にも得られるものである。二次キャッシュ20を追加することにより、上記のキャッシュシステムの効果をさらに増大させることができる。

【0083】実施の形態2。図8は本発明の実施の形態2によるキャッシュシステムのキャッシュ部15Aの構成を示すブロック図である。図8に示すキャッシュ部15Aは、実施の形態2において、図2や図7に示したコンピュータシステムにおいて図1のキャッシュ部15の代わりに用いられるものである。図8のキャッシュ部15AのFSA方式キャッシュテーブルTAは、図1におけるより大きなサイズのアクセスカウン領域ACを有している。具体的には、各キャッシュラインのアクセスカウン領域ACのサイズが例えば16ビットに設定される。この16ビット幅のアクセスカウン領域ACは、例えば実施の形態1の2ビット幅のアクセスカウン領域ACが0から3（十進）までのカウントであったのに比して、0から65535（十進）までをカウントすることができる。

【0084】この実施の形態2においては、前記図示しないアクセスカウンタはすこし異なる方法でアクセスカウン領域ACを管理する。FSA方式キャッシュテーブルTAのあるキャッシュラインがキャッシュヒットした場合、この図示しないアクセスカウンタは実施の形態1と同様に、このキャッシュラインのアクセスカウン値を1だけインクリメントする。しかし、キャッシュ部15Aのミスヒットが発生しても、FSA方式キャッシュテーブルTAの全キャッシュラインのアクセスカウン値の一斉デクリメント（-1）は行わない。従って実施の形態2においては、FSA方式キャッシュテ

ブルTAの各キャッシュラインのアクセスカウン領域ACに格納されるアクセスカウン値はインクリメントされるのみとなる。なお、DM方式キャッシュテーブルTBからFSA方式キャッシュテーブルTAへのデータ転送は実施の形態1と同様に行われる。

【0085】2ビット幅のアクセスカウン領域ACを用いた実施の形態1では、最小のアクセスカウン値（例えば0）を持つキャッシュラインが多くなりがちであるが、この実施の形態2においてはその数を小さくすることができる。従って、実施の形態2のキャッシュシステムによれば、FSA方式キャッシュテーブルTAの高ヒット率（高ヒットカウン）キャッシュライン保持のための工程に、キャッシュラインのヒット数の違いをより直接的に取り込むことができる。

【0086】実施の形態3。図9は本発明の実施の形態3によるキャッシュシステムのキャッシュ部15BにおいてFSA（フルセットアソシアティブ）方式以外のキャッシュテーブルとして用いられる4ウェイセットアソシアティブキャッシュテーブルTCを示す模式図である。前記実施の形態1および実施の形態2では、キャッシュ部（15、15A）のFSA方式以外のキャッシュテーブルとしてDM（ダイレクトマッピング）方式キャッシュテーブルTBを用いたが、この実施の形態3のキャッシュ部15Bは、FSA方式キャッシュテーブルTAと組み合わせられるFSA方式以外のキャッシュテーブルとして、図9に示す4ウェイセットアソシアティブキャッシュテーブルTCを用いる。

【0087】図9を参照すると、この4ウェイセットアソシアティブキャッシュテーブルTCは、各インデックス毎に4本のキャッシュラインを備えている。1つのインデックスに対応する4本のキャッシュラインのそれぞれは、実施の形態1、2におけるDM方式キャッシュテーブルTBの場合と同様に、活性化ビットV2、タグ#2およびデータを格納するための3つの領域を持つ。4本のキャッシュラインからなる行の左端には、LRU（least recently used）値を格納するためのLRU領域が設けられている。このLRU値は、1つのインデックスに対応する4本のキャッシュラインの中から、最近のキャッシュヒットが最も古い、つまり最も昔に使用された（least recently used）キャッシュラインを示すために用いられる。このLRU領域のサイズは、4本のキャッシュラインを表すことが可能なように、例えば2ビットに設定される。

【0088】DM方式キャッシュテーブルTBが1つのインデックスに対して（1つのタグ#2に対応する）1個のデータしか格納できないのと比較して、4ウェイセットアソシアティブキャッシュテーブルTCは1つのインデックスに対して（4つのタグ#2に対応する）4個のデータを格納することができる。従って、4ウェイセ

ットアソシアティブキャッシュテーブルTCのキャッシュヒット率はDM方式キャッシュテーブルTBのキャッシュヒット率より高い。

【0089】以下において、この実施の形態3のキャッシュシステムのキャッシュ部15BにおけるFSA方式キャッシュテーブルTAと4ウェイセットアソシアティブキャッシュテーブルTCの動作について詳細に説明する。なお、4ウェイセットアソシアティブキャッシュテーブルTCにおけるキャッシュヒット判定のためのタグ比較は、DM方式キャッシュテーブルTBの場合と同様に10 入力アドレスデータから抽出されたインデックスとタグ#2とを用いて行われる。4ウェイセットアソシアティブキャッシュテーブルTC（4ウェイセットアソシアティブキャッシュメモリ）の場合は、1つの入力アドレスデータから抽出された1個のインデックスと1個のタグ#2に関して、4つのタグ比較が同時に行われる。実施の形態3のキャッシュシステムの以下の記載以外の動作は実施の形態1のキャッシュシステムの動作とほぼ同じである。

【0090】初期化（コンピュータシステムがリセットまたは電源ONされた際）において、FSA方式キャッシュテーブルTAと4ウェイセットアソシアティブキャッシュテーブルTCのすべてのキャッシュラインが非活性化される。すなわち、全キャッシュラインの活性化ビットV1、V2が「0」にリセットされる。その後、FSA方式キャッシュテーブルTAまたは4ウェイセットアソシアティブキャッシュテーブルTCのあるキャッシュラインにデータが書き込みまたは転送される毎に、そのキャッシュラインが活性化される、すなわちそのキャッシュラインの活性化ビットV1またはV2が「0」から「1」に書き替えられる。 30

【0091】FSA方式キャッシュテーブルTAの各キャッシュラインへのアクセス（キャッシュヒット）の回数（アクセスカウント値）が、各キャッシュラインのアクセスカウント領域ACに格納される。FSA方式キャッシュテーブルTAのあるキャッシュラインにキャッシュヒットが生じた場合、そのキャッシュラインのアクセスカウント値が1だけインクリメントされ、キャッシュ部15Bがミスヒットした場合（またはFSA方式キャッシュテーブルTAがミスヒットした場合）、FSA方式キャッシュテーブルTAの全キャッシュラインのアクセスカウント値が一斉に1だけデクリメントされる。 40

【0092】4ウェイセットアソシアティブキャッシュテーブルTCにインデックスに対応する非活性なキャッシュラインが存在せず、FSA方式キャッシュテーブルTAに1以上の非活性なキャッシュラインが存在する時に、キャッシュ部15Bのミスヒットが発生した場合

（FSA方式キャッシュテーブルTAと、4ウェイセットアソシアティブキャッシュテーブルTCのインデックスに対応した4本のキャッシュラインがミスヒットした 50

場合）には、4ウェイセットアソシアティブキャッシュテーブルTCの4本のミスヒットしたキャッシュラインの内の1つに格納されていたデータ（データとタグ#2）がFSA方式キャッシュテーブルTAの非活性なキャッシュラインのうちの1つに転送される。この際の4本のミスヒットしたキャッシュラインからの選択は、そのインデックスに対応する行のLRU領域に格納されているLRU値に基づいて行われ、LRU値により指定されたキャッシュラインが4ウェイセットアソシアティブキャッシュテーブルTCからFSA方式キャッシュテーブルTAへのデータ転送の転送元として選択される。キャッシュ部15Bのミスヒットによりメインメモリ21（または二次キャッシュ20）からフェッチされたデータは、4ウェイセットアソシアティブキャッシュテーブルTCの（FSA方式キャッシュテーブルTAへデータ（データとタグ#2）が転送された）その選択されたキャッシュラインに書き込まれる。

【0093】4ウェイセットアソシアティブキャッシュテーブルTCにインデックスに対応する非活性なキャッシュラインが存在せず、FSA方式キャッシュテーブルTAが活性なキャッシュラインで一杯の時に、キャッシュ部15Bのミスヒットが発生した場合には、4ウェイセットアソシアティブキャッシュテーブルTCの4本のミスヒットしたキャッシュラインの内の前記LRU値で指定されたキャッシュラインに格納されていたデータ（データとタグ#2）がFSA方式キャッシュテーブルTAの最小アクセスカウント値を持つキャッシュラインのうちの1つに転送される。この際の転送先は、最小アクセスカウント値を持つキャッシュラインのうちからランダムに選択することも可能であるが、最近のキャッシュヒットが最も古いキャッシュラインをその中から選択してそのキャッシュラインを転送先とみなすことも可能である。そのような選択は、例えば、アクセス日時を格納するための領域をFSA方式キャッシュテーブルTAの各キャッシュラインに設けることにより実現できる。キャッシュ部15Bのミスヒットによりメインメモリ21（または二次キャッシュ20）からフェッチされたデータは、4ウェイセットアソシアティブキャッシュテーブルTCの（FSA方式キャッシュテーブルTAへデータ（データとタグ#2）が転送された）前記LRU値で指定されたキャッシュラインに書き込まれる。

【0094】4ウェイセットアソシアティブキャッシュテーブルTCにインデックスに対応する非活性なキャッシュラインが1以上存在する時に、キャッシュ部15Bのミスヒットが発生した場合には、キャッシュ部15Bのミスヒットによりメインメモリ21（または二次キャッシュ20）からフェッチされたデータは、4ウェイセットアソシアティブキャッシュテーブルTCのそのインデックスに対応する非活性なキャッシュラインのうちの1つに書き込まれる。



【0095】上記のように、本発明の実施の形態3によるキャッシュシステムによれば、前記実施の形態1の各効果が得られるほか、4ウェイセットアソシアティブキャッシュテーブルTCの使用によりFSA方式以外のキャッシュテーブルのヒット率をより高めることができ、より高いキャッシュヒット率を持つキャッシュ部15Bを実現することができる。特に、CPU50により枝(branch)の多いプログラムが実行されるような場合に、すなわち、同一のインデックスに対応する多くの命令がCPU50により実行されるような場合に、4

ウェイセットアソシアティブキャッシュテーブルTCのキャッシュヒット率向上の効果が顕著に現れる。従って、実施の形態3のキャッシュシステムは、枝を多く有する複雑なプログラムを実行するようなコンピュータシステムに対して特に有効である。

【0096】なお、この実施の形態3においてFSA方式キャッシュと組み合わせられるFSA方式以外のキャッシュとして4ウェイセットアソシアティブキャッシュを用いたが、FSA方式以外のキャッシュとして、2

ウェイセットアソシアティブキャッシュ、8ウェイセットアソシアティブキャッシュ等を用いることももちろん可能である。

【0097】また、実施の形態2で用いたような大きなサイズのアクセスカウン領域ACを実施の形態3のFSA方式キャッシュテーブルTAの各キャッシュラインに設け、図示しないアクセスカウンタに実施の形態2と同様の動作をさせることも可能であり、その場合は実施の形態2と同様の効果を得ることができる。

【0098】なお、実施の形態1および3で、FSA方式キャッシュテーブルTAの各キャッシュラインのアクセ

スカウン領域ACに格納されるアクセスカウン値の初期値を「0」とであると仮定して説明を行ったが、この初期値は「0」と限るものではない。例えば、アクセスカウン値が「0」に集中することを防ぐためにより大きな初期値を持たせることも可能である。このような場合には、アクセスカウン領域ACのサイズ(ビット数)は少し大きめに設定される。

【0099】

【発明の効果】以上のように、本発明のキャッシュシステムおよびキャッシュ処理方法においては、DM方式キャッシュ(TB)等のFSA方式以外のキャッシュ(TB, TC)と、FSA方式キャッシュ(TA)とが組み合わせられ、キャッシュヒット判定のためのタグ比較が両キャッシュで同時に行われる。FSA方式キャッシュ(TA)の各キャッシュラインに生じたキャッシュヒットの数が図示しないアクセスカウンタによりカウントされ、そのキャッシュヒット数に関するアクセスカウン値が各キャッシュラインのアクセスカウン領域ACに格納される。例えば、FSA方式キャッシュ(TA)のあるキャッシュラインがキャッシュヒットした場合にそ

のキャッシュラインのアクセスカウン値が1ずつインクリメントされ、両キャッシュテーブルがミスヒットした場合にFSA方式キャッシュ(TA)の全キャッシュラインのアクセスカウン値が一斉に1ずつデクリメントされる。

【0100】FSA方式キャッシュ(TA)とFSA方式以外のキャッシュ(TB, TC)とを含むキャッシュシステムのキャッシュヒット率を向上させるために、上記アクセスカウン値を用いて各キャッシュのデータの管理/転送が以下のように行われる。

【0101】FSA方式以外のキャッシュ(TB, TC)にインデックスに対応する非活性なキャッシュラインが存在せずFSA方式キャッシュ(TA)に1以上の非活性なキャッシュラインが存在する時にFSA方式キャッシュ(TA)とFSA方式以外のキャッシュ(TB, TC)の両方がミスヒットした場合は、FSA方式以外のキャッシュ(TB, TC)のミスヒットしたインデックスに対応するあるキャッシュラインに格納されていたデータがFSA方式キャッシュ(TA)の非活性なキャッシュラインのうちの一つに転送され、両方のキャッシュのミスヒットによりCPU等の外部メモリ(20, 21)からフェッチされたデータはFSA方式キャッシュ(TA)へのデータ転送元となったFSA方式以外のキャッシュ(TB, TC)の前記キャッシュラインに書き込まれる。

【0102】FSA方式以外のキャッシュ(TB, TC)にインデックスに対応する非活性なキャッシュラインが存在せずFSA方式キャッシュ(TA)が活性なキャッシュラインで一杯な時にFSA方式キャッシュ(TA)とFSA方式以外のキャッシュ(TB, TC)の両方がミスヒットした場合は、FSA方式以外のキャッシュ(TB, TC)のミスヒットしたインデックスに対応するあるキャッシュラインに格納されていたデータがFSA方式キャッシュ(TA)の最小のキャッシュヒットカウン値を持つキャッシュラインのうちの一つに転送され、両方のキャッシュのミスヒットによりCPU等の外部メモリ(20, 21)からフェッチされたデータはFSA方式キャッシュ(TA)へのデータ転送元となったFSA方式以外のキャッシュ(TB, TC)の前記キャッシュラインに書き込まれる。

【0103】FSA方式以外のキャッシュ(TB, TC)にインデックスに対応する非活性なキャッシュラインが存在する時にFSA方式キャッシュ(TA)とFSA方式以外のキャッシュ(TB, TC)の両方がミスヒットした場合は、両方のキャッシュのミスヒットによりCPU等の外部メモリ(20, 21)からフェッチされたデータはFSA方式以外のキャッシュ(TB, TC)のインデックスに対応する非活性なキャッシュラインに書き込まれる。

【0104】このようなアクセスカウン値(キャッシュ

ヒットカウント値)を用いたデータの管理/転送により、FSA方式キャッシュ(TA)の高キャッシュヒット率のキャッシュラインの保持性能が改善され、キャッシュヒット率が向上される。また、同一のデータのFSA方式キャッシュ(TA)とFSA方式以外のキャッシュ(TB, TC)における重複が完全に防止され、キャッシュシステムのメモリ空間の使用効率が最大まで向上し、キャッシュヒット率の極めて高いレベルへの向上が可能となる。これにより、キャッシュヒット率とデータアクセス速度の向上を極めて高いレベルで実現することができる。

#### 【図面の簡単な説明】

【図1】本発明の実施の形態1によるキャッシュシステムのキャッシュ部15の構成を示すブロック図である。

【図2】本発明の実施の形態1によるキャッシュシステムが適用されたコンピュータシステムの例を示すブロック図である。

【図3】本発明の実施の形態1によるキャッシュシステムの動作を示すフローチャートである。

【図4】図2に示すCPUによって実行されるプログラムの一例を示す模式図である。

【図5】本発明の実施の形態1のキャッシュシステムを用いるCPUが図4のプログラムを2回行う際のアクセス時間を示す模式図であり、(a)は1回目のプログラム実行の場合を、(b)は2回目のプログラム実行の場合を示す。

【図6】図4のプログラムの命令(5)の実行直後の、図1に示すFSA方式キャッシュテーブルTAとDM方式キャッシュテーブルTBの状態を示す模式図である。

【図7】本発明によるキャッシュシステムの他の例をコンピュータシステムに適用した場合を示すブロック図である。

【図8】本発明の実施の形態2によるキャッシュシステムのキャッシュ部の構成を示すブロック図である。 \*

\*【図9】本発明の実施の形態3によるキャッシュシステムのキャッシュ部においてフルセットアソシアティブ方式以外のキャッシュテーブルとして用いられる4ウェイセットアソシアティブキャッシュテーブルTCを示す模式図である。

【図10】典型的な従来のダイレクトマッピング方式キャッシュを示すブロック図である。

【図11】CPUにより実現されるプログラムの例を示す模式図である。

【図12】CPUが図11のプログラムを実行する際の図10のダイレクトマッピング方式キャッシュのキャッシュテーブルの状態変化の例を示す模式図であり、

(a)は命令(1)実行直後の状態を、(b)は命令(5)実行直後の状態を示す。

【図13】従来のダイレクトマッピング方式キャッシュを使用するCPUが図11のプログラムを2回実行する場合のアクセス時間を示す模式図であり、(a)は1回目のプログラム実行の場合、(b)は2回目のプログラム実行の場合を示す。

#### 【符号の説明】

15 キャッシュ部 (一次キャッシュ)

20 二次キャッシュ (外部メモリ)

21 メインメモリ (外部メモリ)

50 CPU (データ処理ユニット)

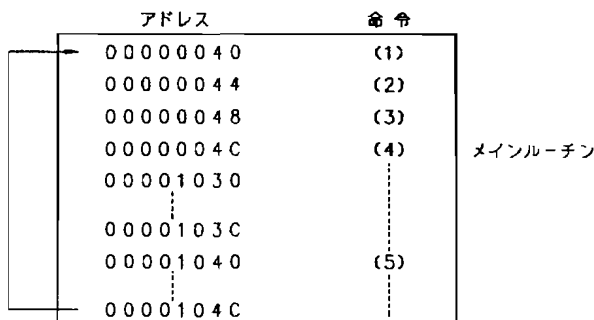
AC アクセスカウンタ領域 (キャッシュヒットカウント格納手段)

TA FSA方式キャッシュテーブル (フルセットアソシアティブ方式キャッシュ)

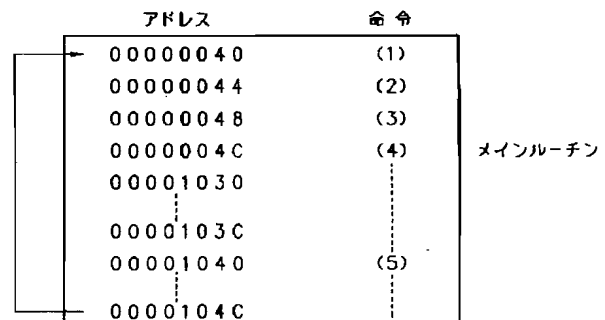
TB DM方式キャッシュテーブル (ダイレクトマッピング方式キャッシュ、フルセットアソシアティブ方式以外のキャッシュ)

TC 4ウェイセットアソシアティブキャッシュテーブル (Nウェイセットアソシアティブキャッシュ、フルセットアソシアティブ方式以外のキャッシュ)

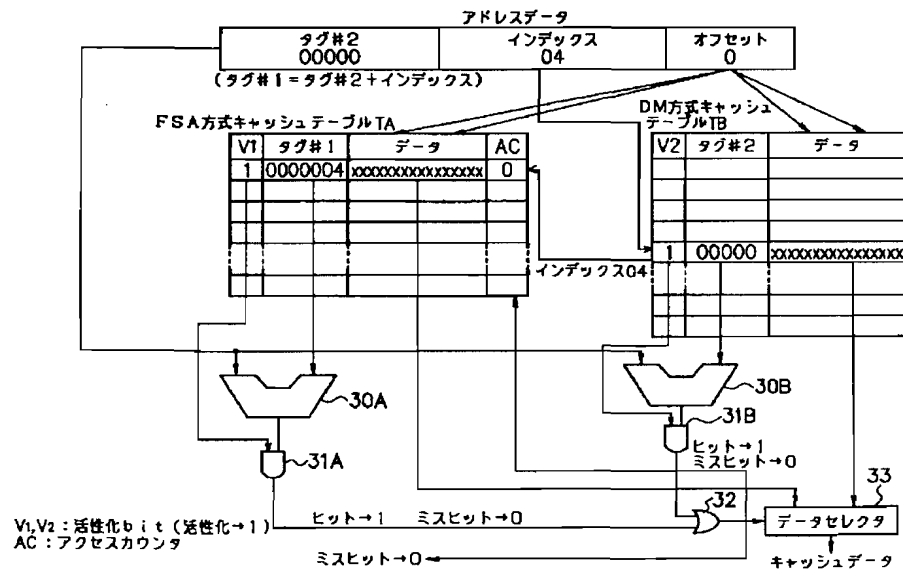
【図4】



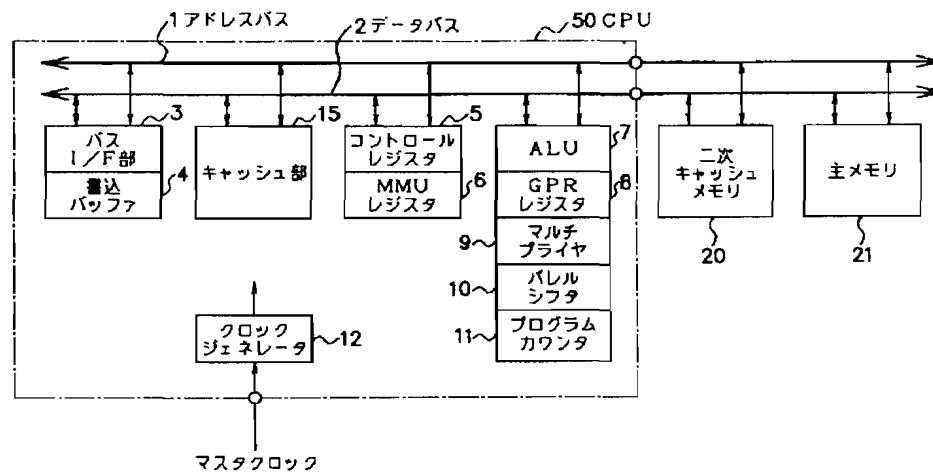
【図11】



【図 1】



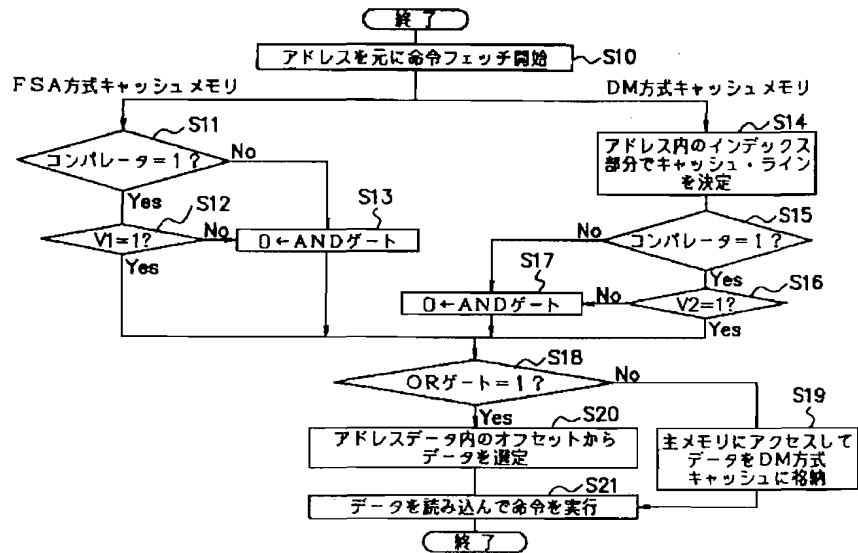
【図 2】



【图 9】

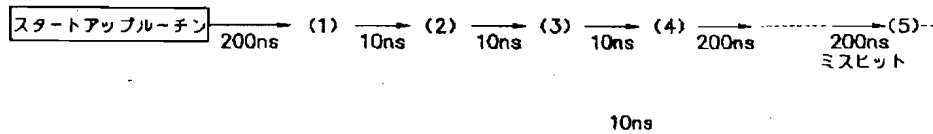
[illegible]

【図3】

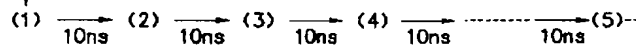


【図5】

(a) 1回目



(b) 2回目以降



主メモリアクセス 200ns=100ns+30ns+30ns+30ns+10ns

キャッシュアクセス 10ns

【図6】

FSA方式キャッシュテーブルA

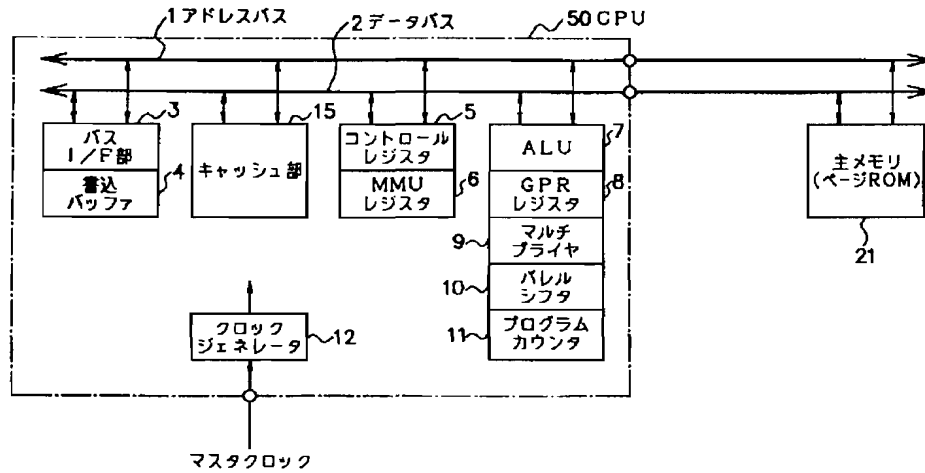
V1	タグ#1	データ	AC
1	00000041	xxxxxxxxxxxxxxxx	0

インデックス00  
インデックス01  
インデックス02  
インデックス03  
インデックス04

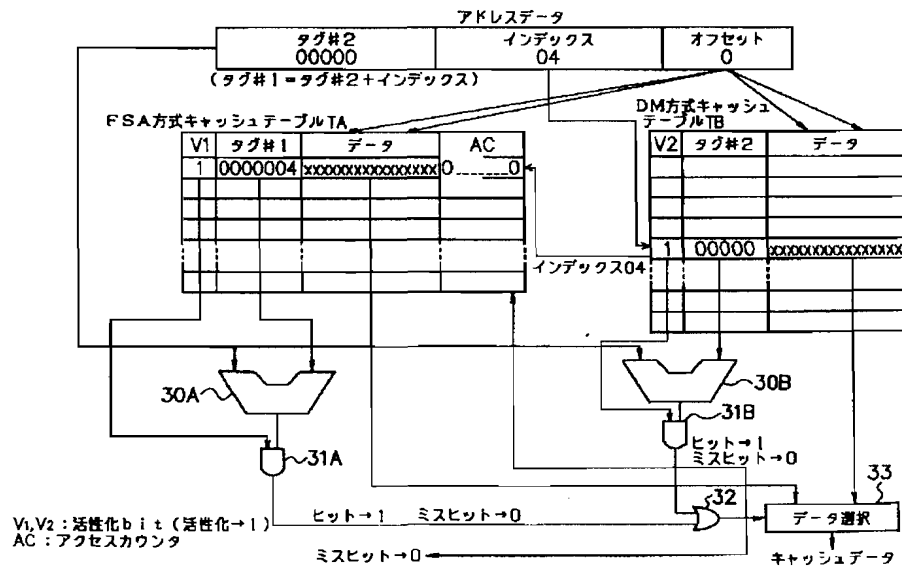
DM方式キャッシュテーブルB

V2	タグ#2	データ
1	xxxxx	xxxxxxxxxxxxxxxx
1	xxxxx	xxxxxxxxxxxxxxxx
1	xxxxx	xxxxxxxxxxxxxxxx
1	xxxxx	xxxxxxxxxxxxxxxx
1	00001	xxxxxxxxxxxxxxxx
1	xxxxx	xxxxxxxxxxxxxxxx
1	xxxxx	xxxxxxxxxxxxxxxx

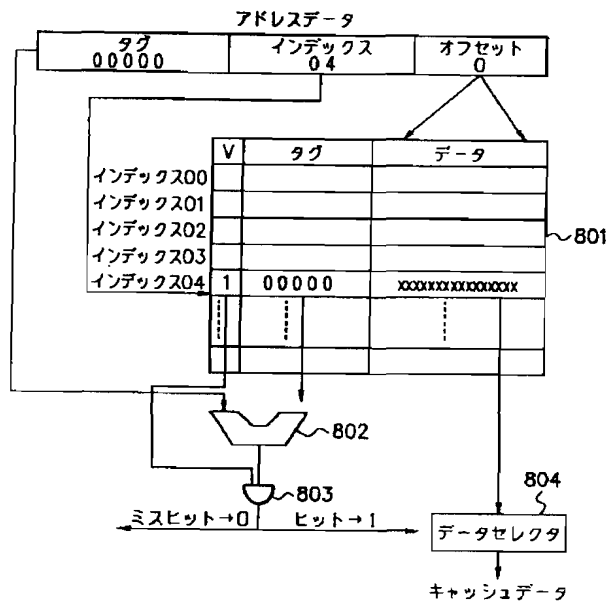
【図7】



【図8】



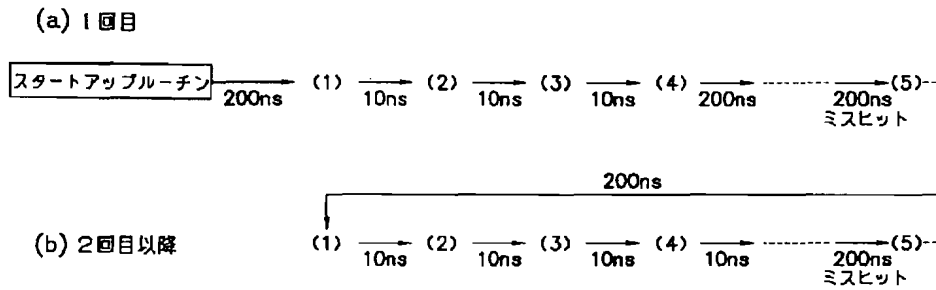
【図10】



【図12】

命令(1) 実行時の ダイレクトマッピング方式キャッシュ				命令(5) 実行時の ダイレクトマッピング方式キャッシュ			
	V	タグ	データ		V	タグ	データ
インデックス00	1	XXXXX	XXXXXXXXXXXXXXXXXXXX	インデックス00	1	XXXXX	XXXXXXXXXXXXXXXXXXXX
インデックス01	1	XXXXX	XXXXXXXXXXXXXXXXXXXX	インデックス01	1	XXXXX	XXXXXXXXXXXXXXXXXXXX
インデックス02	1	XXXXX	XXXXXXXXXXXXXXXXXXXX	インデックス02	1	XXXXX	XXXXXXXXXXXXXXXXXXXX
インデックス03	1	XXXXX	XXXXXXXXXXXXXXXXXXXX	インデックス03	1	XXXXX	XXXXXXXXXXXXXXXXXXXX
インデックス04	1	00000	XXXXXXXXXXXXXXXXXXXX	インデックス04	1	00001	XXXXXXXXXXXXXXXXXXXX
	⋮	⋮	⋮		⋮	⋮	⋮
	1	XXXXX	XXXXXXXXXXXXXXXXXXXX		1	XXXXX	XXXXXXXXXXXXXXXXXXXX
	1	XXXXX	XXXXXXXXXXXXXXXXXXXX		1	XXXXX	XXXXXXXXXXXXXXXXXXXX

【図 13】



主メモリアクセス  $200\text{ns} = 100\text{ns} + 30\text{ns} + 30\text{ns} + 30\text{ns} + 10\text{ns}$   
 キャッシュアクセス  $10\text{ns}$

